VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Gigabitový směrovač na bázi RouterOS

Gigabit router based on RouterOS

2012 **Stanislav Bilík**

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Zadání bakalářské práce

Student: **Stanislav Bilík**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2601R013 Telekomunikační technika

Téma: Gigabitový směrovač na bázi RouterOS
Gigabit Router Based on RouterOS

Zásady pro vypracování:

1. Proveďte analýzu realizace směrovače pod RouterOS s použitím přenosové rychlosti Gb/s.
2. Realizujte gigabitový směrovač na různých HW platformách (powerPC, Mipsbe, Mipsle, x86) a srovnejte jejich výkon (vytížení paměti, procesoru).
3. Na konkrétním příkladu zrealizujte síťovou službu na bázi RouterOS, u nichž je primárně využita přenosová rychlost Gb/s.

Seznam doporučené odborné literatury:

Mikrotik [online]. 2011 [cit. 2011-10-19]. Documentation MikroTik RouterOS Software. Dostupné z WWW: <http://wiki.mikrotik.com/wiki/Main_Page>.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Libor Michalek, Ph.D.**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA * Fakulta elektrotechniky a informatiky * OSTRAVA *

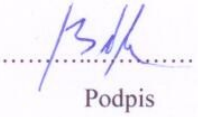
prof. RNDr. Vladimír Vašinek, CSc.
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Dne: 8.2.2012



Podpis

Poděkování

Rád bych poděkoval ing. Liborovu Michalkovi za odbornou pomoc a konzultaci při vytváření této diplomové práce a velmi vstřícné jednání.

Abstrakt

Tato bakalářská práce pojednává o základních charakteristikách a možnostech nastavení operačního systému RouterOS. V tomto textu je pozornost zaměřena na platformu x86 s komunikačním rozhraním 1Gbps. Možnosti využití jsou prezentovány na modelových ukázkách, kdy směrovač zastává funkce NAT a retranslační stanice s integrováním firewallem, autonomním směrováním a dalšími funkcemi, které se v běžné praxi vyskytují. Je zde také nastíněna možnost používání protokolu IPv6 a využití autokonfigurace tohoto protokolu na koncových PC.

Klíčová slova

ACL, Address List, BGP, DHCP, DNS, EoIP, End Station, Filter Rules, Firewall, Gateway, HotSpot, Interface, IP, IPv6, ISP, LAN, Loopback, MAC, MAN, Mangle, MikroTik, NAT, PCI, PPP, PPPoE, PPTP, P2P, Queue Tree, RJ45, RouterOS, Simple Queues, SSH, SuperMicro, Switch, VLAN, VoIP, WAN, Winbox

Abstract

This thesis discusses the basic characteristics and possibilities RouterOS operating system settings. In this text, attention is focused on the x86 platform with communication interface gigabyte. Options are presented using the model shows, which takes the NAT router and relay station integrating firewall, routing and other autonomous functions that occur in everyday practice model with these situations. It also outlines the possibility of using IPv6 autoconfiguration and use this is protocol on the PC.

Key words

ACL, Address List, BGP, DHCP, DNS, EoIP, End Station, Filter Rules, Firewall, Gateway, HotSpot, Interface, IP, IPv6, ISP, LAN, Loopback, MAC, MAN, Mangle, MikroTik, NAT, PCI, PPP, PPPoE, PPTP, P2P, Queue Tree, RJ45, RouterOS, Simple Queues, SSH, SuperMicro, Switch, VLAN, VoIP, WAN, Winbox

Seznam použitých symbolů

Symbol	Jednotky	Význam symbolu
	Mbps	Mega bitů za sekundu
	Gbps	Giga bitů za sekundu

Seznam použitých zkratek

Zkratka	Anglický význam	Český význam
ACL	Access Control List	Seznam pro autoritativní přístup
BGP	Border Gateway Protocol	Protokol autonomního směrování
DHCP	DynamicHhost Configuration Protocol	Protokol pro dynamické hostování
DNS	Domain Name Systém	Systém doménových jmen
EoIP	Ethernet over IP	Ethernetový protokol přes IP
IP	Internet Protokol	Internetový protokol
IPv6	Internet Protokol version 6	Internetový protokol verze 6
ISP	Internet Services Provider	Poskytovatel datových služeb
LAN	Local Area Network	Síť místního rozsahu
MAC	Metropolitan Area Network	Síť městského rozsahu
MAN	Media Access Control	Řízení přístupu k médiu
NAT	Network Address Translation	Síťový překlad adres
PCI	Peripheral Component Interconnect	Sběrnicový systém
PPP	Point to Point Protocol	Protokol spojení bod-bod
PPPoE	Point to Point over Ethernet	Spojení bod-bod přes Ethernet
PPTP	Point to Point Tunneling Protocol	Protokol pro tunelové připojení bod-bod
P2P	Peer to Peer	Síť typu klient - klient
RB	Router Board	Směrovač IP paketu
RJ45	----------	Konektor síťové kabeláže UTP/STP
SSH	Secure Shell	Vzdálené zabezpečené připojení
VLAN	Virtual Local Area Network	Virtuální lokální síť
VoIP	Voice over Internet Protocol	Internetová telefonie
WAN	Wide Area Network	Síť globálního/národního rozsahu

Seznam použitých termínů

Termín	Význam termínu
Daemon	Hardwarová struktura pracující pod OS RouterOS
RouterBoard	Systémová úloha, která je spuštěna na pozadí v operačním systému Linux
Wireless ISP	Poskytovatel datových služeb pomocí bezdrátových technologií

Obsah

Obsah

1 Úvod

Tato bakalářská práce se snaží nastínit základní problematiku, která je každodenně řešena poskytovateli datových služeb, která jsou nezbytně nutná pro hladký a stabilní chod sítě. Před každou praktickou ukázkou jsou v krátkém kontextu shrnuty důvody, proč se daný problém řeší, a kde má řešení největší uplatnění. Nejsou zde zdlouhavé detailní texty, nebo komplexně řešená celá specifická problematika, je to dáno zpravidla rozsáhlostí problému, na který není v této práci místo. Detailní nebo konkrétnější možnosti si čtenář může najít v jiných zdrojích, které se budou zabývat konkrétním problémem. Tato práce má ukázat na problém a navrhnout možné řešení. První kapitola je věnována tématice o základních principech, instalaci a možnostech operačního systému RouterOS, a to včetně krátké historie. Druhá kapitola se zabývá implementací RouterOS na platformu x86. Zároveň se zabývá důvodem výběru této platformy a srovnáním výkonu s jinými platformami. Třetí kapitola je věnovaná tématu NAT. V této tematické kapitole budou popsány možnosti klasického NAT-u 1:1 a NAT-u 1:256. Dále je třetí kapitola je věnována tématice firewall. V této kapitole budou prezentovány možnosti blokace neautorizovaných IP adres, odchytávány potenciálně nebezpečné pakety, či zamezení brutálních útoků na naši technologii. Téma firewall je přesně ukázkovým tématem, které by si zasloužilo vlastní projekt kvůli obsáhlosti. Čtvrtá kapitola nabízí možnosti implementace praktických pravidel z oblastí přidělování šířky pásma a inteligentního přidělování dostupného pásma na retranslační stanici. Dále bude prezentována možnost omezení koncové stanice v komunikaci s okolím dle našeho výběru, nebo přesměrování provozu. Pátá kapitola popisuje další části operačního systému RouterOS, které jsou běžně využívány. Jedná se o témata jako BGP. Pozornost je také věnována tématu IPv6. Toto téma je zastoupeno v závěrečné šesté kapitole. Tato kapitola se zaměřuje na aoutokonfiguraci.

1. MikroTik, RouterOS

Jako začátek této bakalářské práce bude seznámení se s touto platformou. I když název „MikroTik“ či „RouterOS“ není pro mnohé lidi díky hojně využívané technologii Wi-Fi v našem státě cizí, základní seznámení bude na místě.

1.1 Historie

Platforma RouterOS není zdaleka výstřelkem poslední doby, ba naopak prošla si dlouholetým vývojem, který sahá až do zhruba první poloviny osmdesátých let. V té době se v Lotyšsku začal vyvíjet operační systém, který by sloužil jako nástroj pro kvalitní a spolehlivou elektronickou komunikaci armády SSSR. Na vývoji systému pracovalo několik týmů, které byly ale personálně nestabilní, toto snad bylo příčinou, která se projevila důsledkem pravého opaku. MikroTik nebyl univerzální a mnohé funkce v něm byly nestabilní.

Po rozpadu SSSR se od roku 1995 začala tomuto systému věnovat společnost MikroTik (po rozpadu SSSR vznikla z projektu společnost), která vstoupila na trh s jasným cílem vytvářet produkty pro wireless ISP. RouterOS se dočkal značného přepracování, vývojové týmy využily nabyté zkušenosti a vytvořily systém pod názvem MikroTik v2 PC. Tato verze přinesla výraznou stabilitu, příjemnější a rychlejší ovládání, a v neposlední řadě flexibilitu pro různé komunikační rozhraní, podporované tehdejšími standardy na PC. Vývoj jde neustále dopředu, to také můžeme pozorovat i na počtu vydaných verzí. Vezmeme-li v úvahu, že v roce 2007/2008 byla na scéně verze 2.8, a teď na konci roku 2011 můžeme již stahovat stabilní verzi 5.9, je vidět, že za poslední tři roky pokročil skoro o 3 generace. Společnost se snaží implementovat nejnovější technologie a aplikovat je. Proto v tomto operačním systému najdeme funkce od obyčejného NAT serveru, přes firewall, řízení provozu, dynamické směrování, až po MPLS a IPv6. Najdeme ale zde i spoustu dalších funkcí. Tato platforma se postupně dostává z technologie pro last mile i na úroveň systému CORE, kde jsou uplatněny právě gigabitové směrovače. [7]

1.2 Popis platformy RouterOS

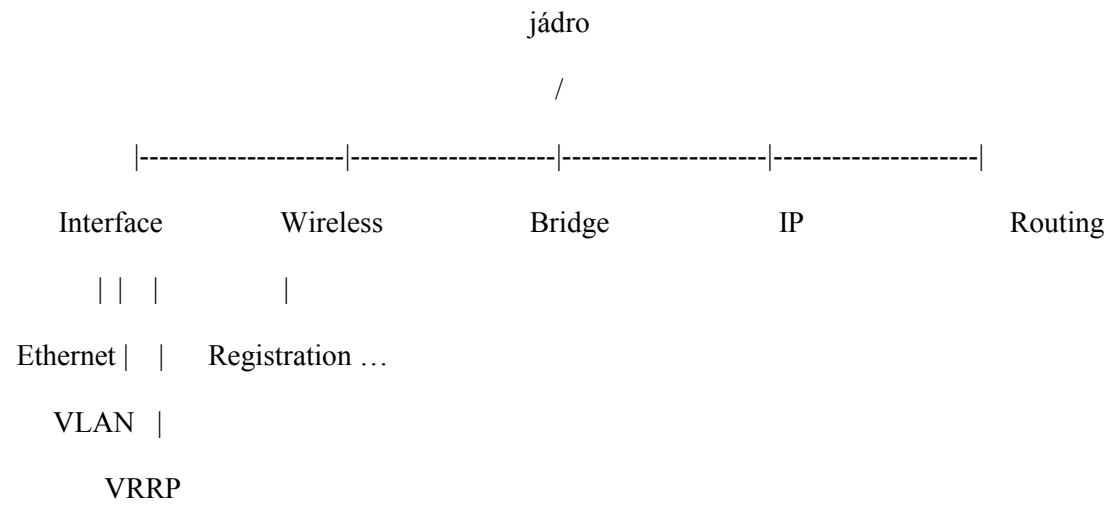
1.2.1 Příkazy

Platforma RouterOS je vytvořena na jádře z Linuxu, avšak práce v terminálu a syntaxe se liší. MikroTik se snaží vyvíjet systém tak, aby byl snáze pochopitelný a intuitivní i pro méně odbornou veřejnost, proto zde nenacházíme žádné příkazy ifconfig či route atd. Abychom mohli něco nastavit, jednoduše zadáme cestu k funkci a napíšeme vlastnost. Veškeré možnosti, které se nám v aktuální úrovni nabízejí lze také zobrazit dvojtým stištěním klávesy TAB. Následující příklad demonstruje přidání IP adresy 192.168.0.1/24 na rozhraní eth0

```
[username@routername] > ip address add address=192.168.0.1/24
interface=eth0
<CR>
```

1.2.2 Balíčky

RouterOS pracuje s tzv. npk balíčky. Každý jednotlivý balíček přináší další specifickou funkci, kterou lze pak v systému využít. Instalace jednotlivých balíčků si můžeme vybrat při instalaci operačního systému, nebo je později doinstalovat samostatně. Samozřejmostí je funkce pro vypínání, nebo zapínání balíčků. Celou strukturu balíčků a zároveň strukturu při práci v terminálu si lze představit jako adresářovou strukturu v Linuxu. Hlavní je jakýsi kořen, což je jádro samotné, a pod ním jednotlivé balíčky, které dále pod sebou mají své specifické funkce, které mají opět své parametry. Tímto způsobem se dostaneme ke každé možné funkci v RouterOS .



1.2.3 Licence

Důležité je podotknout, že některé funkce balíčků, nebo balíčky samotné, lze využít jen ve vyšších úrovních licence. MikroTik jako takový nabízí několik úrovní, rozdíl mezi nimi je především v tom, pro jaký účel je daná implementace určená. Implementace na HW určeného pro příjem služby od ISP k zákazníkovi má logicky nižší licenci než implementace na HW určeného jako retranslační stanici ISP či jiné. Proto například u klienta není možné změnit mód karty ze „station“ na „AP bridge“, atd.

1.2.4 Přístupová rozhraní

RouterOS má pro uživatele k dispozici hned několik přístupových rozhraní. Nejpoužívanějším je program Winbox. Jedná se o software pro platformu Windows, který komunikuje s daemonem integrovaném v jádře systému RouterOS. Ten přenáší patřičné informace z/do operačního systému prostřednictvím TCP/IP. Ojedinělou funkcí mezi směrovači, ale hojně využívanou funkcí, je možnost komunikace Winboxu s RouterOS na 2. síťové vrstvě. Díky tomu je možné vstupovat do systému a udělat nezbytně nutné kroky k tomu, abychom mohli posléze navázat komunikaci na TCP/IP. Dalšími variantami jsou standardní komunikační protokoly, jako je telnet, SSH, SNMP a od verze 4.0 i API. Poslední možností komunikace je nejdůležitější, ačkoliv nejméně využívané rozhraní. Jedná se o konsolový přístup pomocí standardní sériové linky RS232. Nejdůležitější je v případě selhání SW, třeba při upgrade, je to jediná možná cesta přímého přístupu. Z konsole můžeme nastavovat BIOS, takt CPU, boot z eth. nebo NAND atd. Dále lze konzoly využít pro přehrání OS. K tomuto slouží program *Netinstall,* do kterého se nastaví cesta k balíčku *.npk, který se má nainstalovat na RouterBoard.

1.3 Hardware pro implementaci RouterOS

Najít na trhu HW podporující operační systém RouterOS není vůbec těžké. Ba naopak nabízí se hned několik možností, jakými lze tento operační systém nasadit. Když se totiž řekne „MikroTik“, ti znalejší si okamžitě představí malý hardware, na které běží RouterOS. Společnost MikroTik totiž přináší jako řešení nejen operační systém, ale i kompletní hardwarové řešení. Pro výběr produktů z řady MikroTik je jen potřeba úvahy, k jakému účelu nám zařízení bude sloužit, a podle toho vybírat. Společnost MikroTik má ve svém portfoliu několik modelových řad, které se od sebe liší právě účelem, pro který mají být nasazeny. V hardwarové specifikaci jednotlivých modelů je možné si všimnout, že takt CPU jednotlivých hardwarových sestav, se pohybuje od 175MHz CPU až po hodnoty překračující 1GHz, a na platformách x86 i mnohem více. Operační systém RouterOS je

kompilován hned do několika procesorových architektur. Každá modelová řada využívá procesoru takového, který byl uznán jako dostačující pro dané použití. V portfoliu najdeme procesory tipu MIPSle, MIPSbe, PowerPC a kompilaci pro x86. Procesory MIPSle se v dnešní době u RouterBoardu neobjevují, jsou nahrazovány procesory řady MIPSbe. Použití těchto procesorů je převážně u hardwaru určeného pro příjem služby ke klientovi až po malé retranslační stanice. U větších retranslačních stanic, kde je potřeba dosáhnout větší datové propustnosti, a zároveň je větší rozsah směrovacích tabulek, se používá architektury PowerPC. S touto architekturou dosahují RouterBoardy taktu kolem výše uvedeného 1GHz, ale při přetaktování se tak může dostat až na hodnotu 1200MHz. To nám však může ohrozit stabilitu dané hardwarové implementace, proto toto není doporučeno. Všechny výše uvedené architektury mají jedno společné, jsou integrovány na desky malých, až středních rozměrů, zahrnují v sobě vše, co je při práci s touto technologií potřeba. Napájení, které je řešené tak, že je možno využít napájení přes standardní jack i napájení přes PoE. Oba způsoby mají standardně napájecí napětí v rozsahu od 12V-48V, ale existují výjimky. U starších modelů, které ale už dnes na prodejních pultech nejsou k dostání, je třeba způsob napájení definovat nastavením potřebných jumerů, kde se přepíná rozsah napájení a způsob připojení (RB532). Dále malé klientské RouterBoardy zase využívají napájení maximálně do 24V. [6]

Trochu specifickou skupinou je architektura x86. Díky její rozmanitosti je nutné před zakoupením důkladně prověřit, zda je daný chipset na desce podporován. Společnost MirkoTik uvádí na svých stránkách výčet chipsetů, které podporuje, a také čipy jednotlivých periferií, které podporují jejich ovladače. Proto při sestavování směrovače na basi x86 je opravdu potřeba prověřit kompletní sestavu, která má být nasazena, a to včetně periferií. Uklidňující faktor je, že problém s kompatibilitou je vesměs jen u desek, které jsou určeny pro běžné PC, proto je potřeba používat opravdu profesionální HW. Příkladem takového hardwaru může být třeba základní deska SuperMicro PDSMA+ nebo PDSMI, kterou můžeme osadit procesory Intel Xeon. Tyto typy sestav mohou být nasazeny třeba na páteři MAN sítě. Jednotlivá nasazení mohou být ve skupinách s rozdílnými úkoly, (NAT, „malý retrans“, „velký retrans“,…). I při průtoku 80 Mbps a vysíláním této konektivity klientům, kde je použito pro řízení provozu Queue tree, pro směrování BGP, pro ochranu firewall, případně NAT, se při taktu procesoru 4x2.3 GHz nedostane stabilní vytížení ani k 10%, to má příznivý vliv na odezvy, a tak na spojích, které jsou přes tři stanice na 15km vzduchem, nepřekročíme latenci 1-2ms. Samozřejmě při úvaze správně nastavených a synchronizovaných rádiích.

Pro srovnání výkonnosti jednotlivých Hardwarových implementací zde uvádím krátkou tabulku.

Model	Architektura	Takt [GHz]	Paměť RAM [MB]	Propustnost [Mbps]*
RB 411	MIPSbe	300	32	21,81
RB 433	MIPSbe	300	64	21,81
RB 433AH	MIPSbe	680	128	39.68
RB 600	PowerPC	533	64	50.1
RB 800	PowerPC	800	256	66.82
RB 1100AH	PowerPC	1066	2000	184,32
RB 10002xAH	P2020	1066	2000	368,64
RB 1200	PowerPC	1000	512	79.05

Tabulka 1.3 srovnání výkonu RouterBoardu pro RouterOS

*Hodnota je uvedena při zapnutém firewallu, kde RouterBoard pracuje v režimu router a přenáší se 64 bytový rámec. [6]

2 Gigabitový směrovač platformy x86

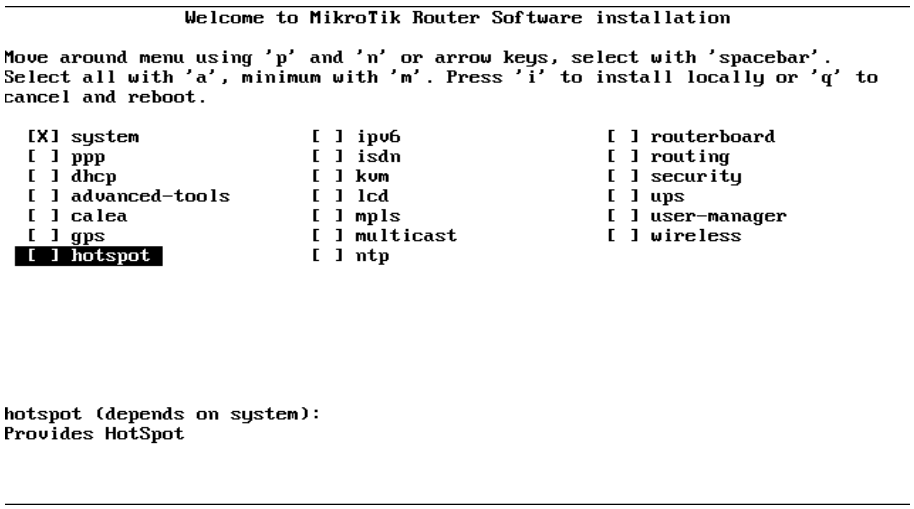
2.1 Výhody architektury x86

Když společnost MikroTik nabízí i hardwarové řešení, naskytuje se otázka, proč tedy nasazovat RouterOS na platformu x86? Vždyť taková sestava je o několik tisícovek dražší, nehledě na to, že musí mít zabezpečen přívod 230V přímo k umístění hardwaru. Nachází-li se hardware někde v těžko dostupném místě, může to být komplikace. Odpověď je jednoduchá. Výkon. Jednoduchá ukázka. Při existenci retranslační stanice, která má distribuovat pomocí bezdrátových technologií kapacitu 150MB k dalším osmi stanicím, které mají být point-to-point, a čtyřem sektorovým anténám, které jsou určené pro koncové vysílání k zákazníkům, dá se lehce odhadnout, že na to ani nejvýkonnější RouterBoard RB 800 nebude bezpečně stačit. Nebude stačit nejen z hlediska toho, že takový počet miniPCI slotu nemá, ale i z hlediska výkonu. Proto je v těchto případech dobré sáhnout po architektuře x86. Koncepce může vypadat takto. Základní deska je osazena redukcemi z PCI na miniPCI, kde jsou radiové karty. Tyto karty jsou spojeny pomocí vlnovodů s anténami. Tímto je dosaženo velmi vysokého výkonu, se kterým je možné disponovat.

Obr. 2.1 Retranslační stanice architektury x86 staršího typu

2.2 Instalace RouterOS na platformu x86

Instalace samotného systému je velice jednoduchá. Na oficiálních stránkách společnosti MikroTik (www.MikroTik.com) jsou ke stažení nejnovější verze RouterOS pro všechny podporované architektury. Pro architekturu x86 je možné v sekci download stáhnout soubor ve formátu iso, které je určené pro vypálení na CD. Po zavedení systému z CD je zobrazena černá obrazovka s možností výběru balíčků, které se mají při instalaci aplikovat. Jednotlivé balíčky je možno také doinstalovat zvlášť pouhým přetažením myši ze složky do winboxu, sekce files. Po vybrání balíčků a stisknutím klávesy „i“ je disk zformátován a zahájená instalace systému. [6]



Obr. 1.2 Instalační obrazovka RouterOS

2.3 Srovnání architektury x86 s architekturou PowerPC

V následující kapitole budou srovnány architektury x86 a PowerPC. Porovnání bude bráno z jednodenního intervalu, kdy budou brány v potaz parametry jako průměrný datový tok skrze směrovač, průměrné využití CPU, průměrné využití paměti RAM a cena. Všechny směrovače byly nasazeny v reálných podmínkách, tj. byly součástí plnohodnotné MAN sítě, kde vykonávaly svou úlohu. Úloha daných směrovačů byla z 90% podobná, kde architektura x86 měla o zbývajících 10% větší zátěž. Jako zátěž je považováno směrování paketů, omezování šířky pásma, počet připojených koncových stanic, datová propustnost atd.

Architektura x86 měla v MAN síti úlohu retranslační stanice. Sestava byla tvořena dvou-jádrovým procesorem Intel Xeon 2GHz, 512MB RAM se základní deskou SuperMicro PDSMA+. Na stanici bylo přihlášeno 43 aktivních koncových stanic a dvě malé retranslační stanice. Stanice byla také součástí dynamického směrování, které zajišťoval protokol BGP, obsah směrovací tabulky činil 335 záznamů. Dále stanice obsahovala 8 bezdrátových rozhraní v módu AP a jedno bezdrátové rozhraní v módu station + nstreme. Stáří cca 4 roky.

Naměřené hodnoty:

Data(IN+OUT):	5,6 Mb/s
Pakety(IN+OUT):	830,5 p/s
Ping:	2,0 ms k této hodnotě připočteme +1, protože se stanice nachází blíže než druhá (o jeden hop a 5 km)
CPU:	14.4%
RAM:	34MB
Cena:	10 000Kč v dnešní době

Architektura PowerPC měla v MAN síti také úlohu retranslační stanice. Sestavu tvořil RouterBoard řady RB 800, což je nejvýkonnější řada RouterBoardu s porty miniPCI, které jsou potřeba pro bezdrátové rozhraní. Na stanici bylo obsluhováno 51 aktivních koncových stanic. Stanice byla také součástí dynamického směrování pod protokolem BGP. Směrovací tabulka obsahovala 431 záznamů. Na stanici bylo realizováno 6 bezdrátových rozhraní v režimu AP a jedno bezdrátové rozhraní v režimu station. Staří stanice je do 1,5 roku.

Naměřené hodnoty:

Data(IN+OUT):	3,9 Mb/s
Pakety(IN+OUT):	583 p/s
Ping:	5,2 ms k této hodnotě připočteme +1, protože se stanice nachází blíže než druhá (o jeden hop a 5 km)
CPU:	31.8%
RAM:	32MB
Cena:	6500Kč

Z uvedených dat je možné vidět, že architektura x86 je výkonnější než RB 800. I když je uvedená sestava architektury x86 podstatně starší, nebyla překonána ani nejnovějšími RouterBoardy. Dále při pohledu na naměřená data se může zdát, že RB 800 alespoň x86 částečně dohání. Pravdou však je, že x86 nebyla ani zdaleka na hranicích svých možností. Kdyby byl generován daleko větší objem dat, určitě by bylo dosaženo hranice propustnosti dosahující 280 Mb/s minimálně. Tato je daleko za hranicí RB 800. Z těchto údajů vyplývá, že v případě výstavby retranslační stanice na páteři MAN sítě se určitě vyplatí využít architektury x86, a to i přes větší cenu, neboť jen tak dosáhneme dobrých latencí a vlastností sítě. RouterBoardy typu RB 800 se hodí jako koncové retranslační stanice většího typu. Grafy tohoto měření jsou uvedeny v příloze C.

2.4 Srovnání architektury x86 s architekturou MIPSbe/le

Ačkoliv se to zdá nevhodné, budou v této kapitole popsány gigabitové směrovače architektury MIPS. Skupina, kterou takovéto směrovače tvoří, se hodí nejlépe pro domácí nebo kancelářské použití. Jejich návrh počítal právě s nasazením do takovýchto podmínek, takže tomu odpovídá i výkon. Na druhou stranu se dá říci, že mezi platformami, které jsou na trhu ve stejné kategorii dostupné, se jedná o výhodný poměr (výkon + funkce)/cena. Srovnání architektur MIPSbe/MIPSle s architekturou x86 je spíše jen demonstrativní. Testování proběhlo opět na zařízeních, která jsou reálně nasazená do provozu na síti. Stanice však nemohly ani zdaleka vykonávat podobné úlohy jako x86, neboť jejich charakteristiky jsou velice rozdílné. Grafy tohoto měření jsou uvedeny v příloze D.

Architekturu MIPSbe tvoří RB 750G. Tato architektura byla nasazena jako koncový router pro úlohu firewall. Na prvním ethernetovém rozhraní byl aplikován příjem datových služeb (WAN). Zbývající čtyři ethernetové rozhraní sloužily pro připojení 20 koncových PC stanic (LAN). Pro autorizaci a přístup na síť byla na rozhraních LAN aktivována funkce HotSpot. Vedle této autorizace bylo prováděno přidělování šířky pásma pomocí Queue tree a také překlad adres. Dále byly nastaveny běžné služby, jako SNTP atd.

Naměřené hodnoty:

Data(IN+OUT):	1,3 Mb/s
Pakety(IN+OUT):	125.1 p/s
Ping:	17,4 ms
CPU:	11.1%
RAM:	20MB
Cena:	1200Kč

Další architekturou, na které bylo provedeno měření, je MIPSle. Jedná se o historicky nejstarší architekturu, která ve své HW implementaci nemá gigabitový ethernet. Tato architektura už není ani nabízena v protfilu společnosti MikroTik, a proto se dá toto měření brát jen jako měření pro přehled. Test byl konkrétně prováděn na hardwarové implementaci RB532, která sloužila jako koncová retranslační stanice. Tato stanice obsluhovala 11 bezdrátových koncových stanic na třech rozhraních v módu APbridge. Jedno rozhraní v módu station sloužilo pro příjem datové služby. Retranslační stanice byla součástí dynamického směrování BGP, při čemž směrovací tabulka obsahovala 395 záznamů. Autorizace byla realizována pomocí MAC filtru.

Naměřené hodnoty:

Data(IN+OUT):	0,392 Mb/s
Pakety(IN+OUT):	66,4 p/s
Ping:	92.7 ms
CPU:	13.8%
RAM:	18MB
Cena:	500-600Kč – pouze bazarové kusy

Při pohledu na naměřená data je vidět, že architektura MIPSle, konkrétně RB 532, je pro provoz na síti už pomalu nedostačujícím prvkem. Nároky kladené na výkon a propustnost již přesáhly možnosti, které nám tato implementace nabízí. Nasvědčují tomu i naměřená data. Je vidět, že i přes relativně malý datový a paketový průtok stanice využívá stejných prostředků jako retranslační stanice na architektuře x86 s průtokem 5,6 MB/s. Dále se dá říci, že při pohledu na naměřené hodnoty je jednoznačně vidět vývoj a posun výkonu vpřed v různých HW implementacích. Také je dobře vidět poměr ceny a výkonu. Z toho vyplývá, že pro každé nasazení je důležité opravdu vědět kam a za jakým účelem je daná implementace nasazována, a to včetně budoucího vývoje. Pro každé měření jsou vypracovány grafy, které jsou uvedeny jako příloha této práce.
[3]

3 Gigabitový směrovač v roli NAT, Firewall, Access gateway

Jedná z možností, která se skoro sama nabízí, je využití RouterOS do role NAT s integrovaným firewallem. Díky snadného a přehledného uživatelského rozhraní, které Winbox nabízí, je možné efektivně vytvářet pravidla překládání adres a filtrace. Tato nastavení pak chrání síť před útoky, nebo jinými nechtěnými jevy z oblasti bezpečnosti sítě. Hardwarová konfigurace pro tuto roli by měla mít opravdu co největší propustnosti, a zohledňovat potřebu dostatečného výpočetního výkonu, který je potřeba pro bleskový překlad adres. Záměrem je překládat adresy všech klientů z privátního rozsahu do veřejného. Zákazníkům je poskytovaná překládaná veřejná adresa, což znamená, že se po MAN síti pohybuje prostřednictvím přidělené lokální adresy a až na samotném konci je adresa přeložena 1:1. Toto řešení má své výhody i nevýhody. Výhodou je případná restrukturalizace veřejného rozsahu adres, v takovém případě se změní pouze údaj v pravidle. V opačném případě by bylo nutné navštívit i tu nejvzdálenější retranslační stanici, aby jí byla změněna IP adresa. Naopak nevýhoda je, že mnohé technologie využívají přímého spojení. Jedná se například o VoIP, nebo platební terminály bank, které mají s překladem problém, a je nutné pro daný incident pak navrhnout jiné řešení. Část realizace NATu je zahrnuta do praktické ukázky, kde jsou popsána různá nastavení a role, které by mohl server NAT na hranicích MAN<-> WAN sítí obsahovat.

3.1 NAT v praxi

Následující ukázka předpokládá fakt, že má být do vznikající sítě postaven server NAT, který bude obsluhovat níže popsané služby. Server se bude nacházet na samé hranici MAN sítě a bude tvořit jediný přístupový bod do sítě providera (není akceptována situace, že existuje na síti záložní konektivita). Dále na tomto serveru bude potřeba aplikovat pravidla pro přístup z WAN sítě, respektive pravidla pro odchozí spojení z MAN. Dále server bude obsluhovat služby pro PPP. Jelikož bude zapotřebí v ukázce připojit i vzdálený uzel, bude pozornost věnována vytvoření EoIP tunelu. Dále se tato kapitola zabývá nasazením VLAN. Je to dáno tím, že platforma x86 je standardně vybavena pouze se dvěma, ojediněle se čtyřmi, porty rozhraní RJ45, což vždy nemusí pro konkrétní řešení stačit. [6,1]

3.2 Nastavení VLAN

Jako první si v našem serveru připravíme všechny VLAN-y, které budeme pro komunikaci potřebovat. Podle uvedeného schématu topologie v příloze budeme potřebovat nastavit 8 VLAN. Nastavení VLAN v systému RouterOS je velice jednoduché, stačí zadat 3 základní parametry pro nastavení plně funkční VLAN-y. Jedná se o parametry Name, VLAN ID, Interface. Funkci ARP doporučuji nechat zapnutou.

Při tvoření VLAN je dobré mít stanovená obecná pravidla, podle jakých budou VLAN-y tvořené. Číslování může mít různá kritéria, nebo může být odvozeno od různých vlastností. V této BP je počítáno s tím, že VLAN ID je odvozeno od čísla portu, ve kterém je zařízení ve switchi zapojeno. Toto pravidlo má ale jeden nedostatek. Jedná se o počátek číslování, které musí začít od VLAN ID 11. Je to dáno tím, že pro první port ve switchi by musel být zvolený VLAN ID 1, tato hodnota je však reservována jako default ID u L2 switchů a dalších zařízení používající VLAN. Z těchto důvodu není možné číslovat od VLAN ID =1. Zkrátka pro ethX platí, že

$$x = VLAN\_ID\ -\ 10 \tag{3.1}$$

Příkaz pro založení VLAN:

```
/interface vlan

add arp=enabled disabled=no interface=ether2lanradia mtu=1500 \
    name=vlan11 use-service-tag=no vlan-id=11

add arp=enabled disabled=no interface=ether2lanradia mtu=1500 \
    name=vlan12 use-service-tag=no vlan-id=12
```

Obdobně budou založeny i VLAN-y 13-18. Po zadání těchto zbývajících parametrů jsou potřebné VLAN-y založeny. Tímto jsou nastavena potřebná rozhraní a je možné k nim přidělovat IP adresy.[1]

3.3 Vytvoření tunelu EoIP

V sítích je běžnou záležitostí tunelové připojení pomocí různých protokolů pro tunelovaná připojení. Důvodů, proč po takovém řešení sáhnout, může být hned několik. V každém případě je však nutné zvolit ten nejvhodnější pro danou situaci. Rozhodovacími faktory může být třeba typ serveru, nebo zprostředkovávané služby, a jeho umístění. V tomto případě je zvolen EoIP tunel, protože

nejvěrohodněji simuluje fyzické připojení média, což má značné výhody. Nevýhodou je, že se zmenšuje MTU, což pro velké rámce dat není dobré. V následujícím případě to ale vadit nebude, protože je stanoveno, že se jedná o malý poštovní server, jehož majitel si platí SPAM filtr, který se realizuje na jednom ze serveru v DMZ zóně. Klient se nachází mimo MAN síť. Modelová situace má následující parametry.

Adresa MikroTiku klienta (realizuje firewall před samotnou LAN sítí klienta) je: 62.168.51.150

Adresa klienta určena pro lokální komunikaci (v rámci hostící EoIP tunelu): 172.30.5.13/30

Adresa NAT serveru: 93.99.109.1

Adresa pro lokální komunikaci: 172.30.5.14/30

V prvním kroku je vytvořeno nové rozhraní EoIP. Jediný podstatný parametr pro vytvoření je adresa protistrany. Tato adresa musí být veřejná, pokud se protistrana nachází mimo naši MAN síť. V případě nutnosti implementovat více EoIP tunelů je nutné, dodat i tunel ID, které musí být na obou stranách stejné.

```
/interface eoip
add arp=enabled disabled=no l2mtu=65535 local-address=0.0.0.0 \
    mac-address=02:4F:38:5B:C0:6C mtu=1500 name=eoip_server \
    remote-address=62.168.51.150 tunnel-id=10
```

Následuje přidání IP adresy na rozhraní EoIP tunelu.

```
add address=172.30.5.14/30 comment=lublice disabled=no \
    interface=eoip_server network=172.30.5.12
```

analogicky je postupováno i při vytváření tunel, a zadávaní adresy na straně klienta.

3.4 Přiřazení IP adres rozhraním

Jeden z rozsahů, který byl obdržen od ISP je 213.192.39.0/24 bude použit jako veřejný rozsah pro překládání privátních adres a druhý rozsah 93.99.109.0/24 pro technologii. Transportní adresa ISP je 213.192.3.120/30 na rozhraní eth1WAN. Jako první bude přiřazena a založena IP adresa serveru NAT, kterou se bude v síti prezentovat. Tato adresa bude na rozhraní loopback, které bude vytvořeno přidáním rozhraní bridge, ale nebude mu přiřazen žádný port. Takto vytvořena adresa má výhody z hlediska autonomního směrování a redundantních tras, a to protože není vázaná na žádné fyzické rozhraní, ale pouze na virtuální. Účel každé přidané adresy je pro přehled definován v parametru "comment", který je součástí v příkazu.

3.4.1 Vytvoření rozhraní loopback:

```
 /interface bridge
add admin-mac=00:00:00:00:00:00 ageing-time=5m arp=enabled \
auto-mac=yes disabled=no forward-delay=15s l2mtu=65535     \
max-message-age=20s mtu=1500 name=loopback priority=0x8000 \
protocol-mode=rstp transmit-hold-count=6
```

3.4.2 Zakládaní IP adres:

```
/ip address
add address=213.192.3.122/30 comment="transportni subnet WAN" \
disabled=no interface=ether1wan network=213.192.3.120

add address=93.99.109.57/30 comment="transport retrans 1" \
disabled=no interface=vlan14 network=93.99.109.56

add address=93.99.109.49/30 comment="transport retrans2" \
disabled=no interface=vlan15 network=93.99.109.48

add address=93.99.109.61/30 comment="transport retrans3" \
disabled=no interface=vlan16 network=93.99.109.60

add address=93.99.109.1/32 comment=loopback disabled=no \
interface=loopback network=93.99.109.1
```

Je vidět, že zakládání IP adres není opět nic složitého. Rozsah 213.192.39.0/24 nebude přiřazen žádnému rozhraní, protože překlad bude probíhat při průchodu paketu. Nikoliv na vstupu, nebo výstupu na daném rozhraní.

3.5 Překlad adres (NAT)

Nyní je možné vytvářet pravidla pro překlad, která uplatníme na síti. V této demonstraci bude využito v zásadě pouze dvou druhu překladu. A to překlad 1:1 a 1:256.

Překlad 1:1 je takový překlad adres, kdy je jedné privátní adrese přidělená jedna veřejná. Může se jednat o případy, kdy si klient přeje mít přiřazenou veřejnou, třeba za účelem sledování domácích kamer. Jedná se o překládání cílové i zdrojové adresy.

Překlad 1:256 je takový překlad adres, kdy je jedné veřejné adrese přiřazeno 256 adres (jedno C). Taková situace vznikne na retranslační stanici pracující v módu point-to-multipoint, kdy na jedno rozhraní je připojeno několik koncových stanic, na které vedou spojovací rozsahy 172.20.AAA.BBB/29. Takto každá koncová stanice dostane vlastní síť, ale jako celek jsou pod jednou veřejnou. V takovém případě můžeme pro ně zavést reverzní záznam třeba ve tvaru cust1.retrans2.provider.cz V tomto případě se jedná jen pouze o překlad na základě zdrojové adresy, tedy koncové stanice.

3.5.1 Překlad 1:1 (NAT)

V tomto případě bude potřeba překládat adresu serveru, který je připojen přes EoIP tunel.

Překlad z privátní adresy na veřejnou

```
/ip firewall nat
add action=src-nat chain=srcnat comment="NAT 1:1 server na EoIP" \
disabled=no out-interface=ether1wansloane src-address=172.30.5.13 \
to-addresses=213.192.40.25
```

Zpětný překlad veřejné adresy na privátní

```
/ ip firewall nat
add action=dst-nat chain=dstnat disabled=no \
dst-address=213.192.39.25 to-addresses=172.30.5.13
```

3.5.2 Překlad 1:256

Zde je možné vidět, že každá retranslační stanice má svůj rozsah pro koncové stanice, které jsou k ní připojeny. Každý takový rozsah je překládán na jednu veřejnou.

```
/ip firewall nat
add action=src-nat chain=srcnat comment=\
"Prenatuj vnitrni sit na realnou IP 213.192.39.1 - retrans1" \
 disabled=no out-interface=ether1wansloane \
 src-address=172.20.80.0/24 to-addresses=213.192.39.1

add action=src-nat chain=srcnat comment=\
    "Prenatuj vnitrni sit na realnou IP 213.192.39.2 - retrans2" \
     disabled=no out-interface=ether1wansloane \
     src-address=172.20.90.0/24 to-addresses=213.192.39.2

add action=src-nat chain=srcnat comment=\
    "Prenatuj vnitrni sit na realnou IP 213.192.39.3 - retrans3" \
    disabled=no out-interface=ether1wansloane \
    src-address=172.20.121.0/24 to-addresses=213.192.39.3

add action=src-nat chain=srcnat comment=\
    "Prenatuj vnitrni sit na realnou IP 213.192.39.4 - retrans4" \
    disabled=no out-interface=ether1wansloane\
    src-address=172.20.126.0/24 to-addresses=213.192.39.4
```

Tímto byly zavedeny základní překládací pravidla a jsou plně schopná provozu. Samozřejmostí je, že MikroTik nabízí daleko širší škálu možností překladu, které se můžou v praktické implementaci, nebo dozoru sítě hodit. Jeden příklad za všechny.[6]

3.5.3 Přesměrování provozu

V každé síti se naskytne situace, kdy je v rámci zlepšování kvality služeb nutné uplatnit nějakou restrukturalizaci. Takovéto změny na síti se však mohou v zásadním případě dotknout i koncových stanic klientů. V následujícím modelovém příkladu bylo nutné změnit DNS server z 62.240.161.226 na 213.192.40.6.

I když bylo vynaloženo veškeré úsilí pro předání informace o změně koncovému uživateli a zároveň byly provedeny změny techniky ISP, nemůžeme zaručit, že bylo změněno vše. V takovém případě jsou tři možnosti.

a) Buď bude vytvořeno pravidlo, že jakákoliv adresa, která žádá o komunikaci se starou DNS adresou, bude přesměrována na nový DNS server.
b) Veškerý provoz na starý DNS server se bude logovat do Address Listu a dané koncové stanice, které se nám zachytí, navštívíme.
c) Uplatníme obě pravidla

Nejlogičtěji připadá v úvahu bod „c“. V tomto případě existuje možnost vidět, kdo nemá změnu provedenou a zároveň daná stanice nebude v komunikaci omezená. Nastavení takovéhoto pravidla je následující.

Jako první bude vytvořeno pravidlo pro zachytávání všech, kdo chtějí komunikovat se špatným DNS.

```
/ip firewall nat
add action=add-src-to-address-list address-list="stanice bez DNS" \
    address-list-timeout=5d chain=dstnat comment=\
    "uloz klienta se spatnymi DNS" disabled=no
    dst-address=62.240.161.226 dst-port=53 protocol=udp
```

Následující pravidlo přesměruje provoz na správný DNS server. Pravidlo pracuje se seznamem, který byl vytvořen pravidlem výše. Tudíž i posloupnost pravidel musí být dodržena, a to první zachytávání, pak přesměrováni!

```
/ip firewall nat
add action=src-nat chain=srcnat disabled=no dst-port=53 protocol=udp
    src-address-list="stanice bez DNS" to-addresses=213.192.40.6 \
    to-ports=53
```

3.6 Firewall

Problematika funkce firewall by nejlépe zasloužila vlastní bakalářskou práci. Díky její obsáhlosti a spoustou možností, jakými lze firawall nastavovat, je firawall velice mocným nástrojem. Firewallem je možné řídit provoz, definovat pravidla provozu, zachytávat nežádoucí spojení, nastavovat limity a podobně. Výklad všech funkcí nebude v této práci možný, budou však uvedena příkladová pravidla, která by na serveru NAT mohla být uplatněna.

MikroTik používá standardně tři různé základní podmínky, které definují, kdy se pravidlo má uplatnit. Jedná se o podmínky „input“, „output“, „forward“. Jejich podmínka uplatnění je zřejmá už z názvu, „při vstupu“, „pří výstupu“, „při průchodu“.

Input – uplatní se v tu chvíli, vstupuje-li paket na libovolné rozhraní, určený pro libovolnou adresu

Output – uplatní se v tu chvíli, opouští-li paket libovolné rozhraní serveru, nesoucí libovolnou zdrojovou adresu

Forward – uplatní se v tu chvíli, prochází-li pouze paket skrz server a nikterak do něho nemusí zasahovat kromě směrování. Důležitou poznámkou je, že se pravidla aplikují od těch nejvýše umístěných (#0, #1..) až k těm nejníže umístěným. Bude-li zablokován provoz už v druhém pravidle, není možné ho v třetím pravidle filtrovat a podobně [1,6]

3.6.1 Vytvoření praktických pravidel

Pro praktickou demonstraci firewallu zde budou uvedena pravidla pro následující případy.

- Blokování stanic, které rozesílají SPAM
- Blokování brutálních útoků na ssh
- Blokování známých virů, jako je Message Worm, Blaster Worm, Doom

3.6.1.1 Blokování stanic, které rozesílají SPAM

Jistě každý ISP bude souhlasit, že kontrola SPAMu je v dnešní době dost žhavé téma. Je odhadováno, že SPAM dělá 81% emailového provozu na internetu, což je až kritická situace. Z tohoto důvodu není od věci aplikovat filtry, které by omezily rozesílaní SPAMU z přiděleného adresního rozsahu sníží se tak tím provoz na trasách sítě. Hlavním důvodem však je, aby přidělené adresy nebyly ve světě označené jako adresy rozesílající SPAM, a tím nebyly zařazeny do světových Blacklistů. Kdyby se totiž tak stalo, značně by to mohlo zkomplikovat emajlovou komunikaci. Tato situace platí dvojnásob, je-i na dané adrese provozován poštovní server, a to ať už od ISP, nebo od klienta. Dostat se totiž z takovéhoto seznamu, není vůbec jednoduché, může to trvat i v řádech několika měsíců, u extrému 2-5 let. To už je lepší danou adresu vyřadit a zavést novou, protože je zaručeno, že z blokované adresy nebude nic odesíláno a její odstranění z blacklistu je o něco lehčí.

Pravidlo pro SPAM se skládá z několika pravidel, která se postupně uplatňují. Jelikož bylo řečeno, že pravidla se uplatňují odshora dolů, bude první pravidlo porovnávat IP z address listem „spamera“. Pravidlo se dá slovy vyjádřit následně.

„Je-li zdrojová adresa zařazena v Address Listu jako spamer, potom zablokuj“

Následující pravidlo bude testovat adresní rozsah koncových stanic na počet spojení a aktuální průtok. Dosáhne-li nastaveného max. limitu, bude zařazen jako spamer do Address Listu. Testovat se budou rozsahy 172.20.0.0/16 a 213.192.39.0/24.

```
add action=drop chain=forward comment=\
    "BLOCK SPAMMERS OR INFECTED USERS" disabled=no dst-address=\
    !77.48.46.128/26 dst-port=25 protocol=tcp \
    src-address-list=spammer
add action=add-src-to-address-list address-list=spammer \
    address- list-timeout= \
    1d chain=forward connection-limit=30,32 disabled=no \
    dst-port=25 limit=50,5 protocol=tcp src-address=172.20.0.0/16
add action=add-src-to-address-list address-list=spammer \
    address- list-timeout= \
    1d chain=forward connection-limit=30,32 disabled=no \
    dst-port=25 limit=50,5 protocol=tcp src-address=213.192.39.0/24
```

3.6.1.2 Blokování brutálních útoků na ssh

Dalším často viditelným problémem jsou takzvané „Brutální útoky“. Tento typ útoků napadá komunikační rozhraní serveru, směrovačů a dalších zařízení s cílem získat přístup do daného zařízení. Brutální útoky mohou být tvořeny na základě word listu, který obsahuje pravděpodobná možná hesla a hesla, která byla již použitá a zjištěná jinde na serverech. Nepodaří-li se díky tomuto listu získat přístup, je automaticky v kombinacích používaná abeceda znaků, nad kterou jsou tvořeny různé možnosti, kde každá vytvořená možnost je vyzkoušena. Podle rozsahu abecedy se exponenciálně zvětšuje počet kombinací, proto se doporučuje volit taková hesla, která zahrnují různé znaky (písmeno, číslo, speciální znak).

Uvedený filtr má tři úrovně a skládá se z šesti pravidel. Jako první pravidlo je opět zablokování na základě address listu, kde jsou uložené zdrojové ip adresy s označením „ssh attack“. Následující pravidlo zkoumá, zda zdrojová adresa, která žádá spojení ssh, již není v address listu jako level3. Pokud ano, je označena jako ssh attack.

Další pravidlo zkoumá, zda zdrojová adresa, která žádá spojení ssh, již není v address listu jako level2. Pokud ano, je označena jako level3.

Čtvrté pravidlo opět analogicky zkoumá, zda zdrojová adresa, která žádá spojení ssh, již není v address listu jako level1. Pokud ano, je označena jako level2.

A konečně poslední pravidlo. Zdrojová adresa, která žádá spojení ssh, je automaticky zařazena do address listu jako level1.

Všechna pravidla zkoumající přítomnost zdrojové adresy v address listu, kromě blokovacího, mají dobu zapsaní v address listu 1 minutu, dohromady tedy jsou k dispozici 3 minuty pro zkoušení přihlášení, což je pro přístup člověka, který má adekvátní přístup do serveru, více než dostačující.

```
add action=drop chain=input comment="drop ssh brute forcers"
    disabled=no dst-port=22 protocol=tcp \
    src-address-list=ssh_attack
add action=add-src-to-address-list address-list=ssh_attack \
    address-list-timeout=1w3d chain=input \
    connection-state=new disabled=no \
    dst-port=22 protocol=tcp src-address-list=level3
add action=add-src-to-address-list address-list=level3 \
    address-list-timeout=1m chain=input \
    connection-state=new disabled=no \
    dst-port=22 protocol=tcp src-address-list=level2
add action=add-src-to-address-list address-list=level2 \
    address-list-timeout=1m chain=input \
    connection-state=new disabled=no \
    dst-port=22 protocol=tcp src-address-list=level1
add action=add-src-to-address-list address-list=level1 \
    address-list-timeout=1m chain=input \
    connection-state=new disabled=no \
    dst-port=22 protocol=tcp src-address-list=!naseservery
```

Nemělo by být opomenuto, že útočník, který se snažil nabourat do serveru NAT, to mohl zkusit i jinde na síti a mohlo se mu to třeba povést. Jako obrana je na místě založit úplně poslední pravidlo, které zablokuje již navázanou komunikaci s IP adresou, která se dostala do Address listu jako ssh_atack.

```
add action=drop chain=forward comment="drop ssh brute downstream"
    disabled=no dst-port=22 protocol=tcp src-address-list=sh_attack
```

[5,6]

3.6.1.3 Blokování známých virů jako je Message Worm, Blaster Worm, Doom

Další nepříjemnou skutečností je, že se po internetu šíří spousta škodlivého softwaru, který působí mnohé nepříjemnosti. Problematiku virů není třeba značně připomínat, jelikož se každý v praxi s nějakým tím virem jistě potkal. V této sadě pravidel budou blokovány vybrané viry.

Celý princip je velice jednoduchý. Mezi předchozí sadu pravidel, tj. mezi spamery a ssh_attack, bude vloženo pravidlo, které přeskočí na dané místo, a testování bude pokračovat z toho místa. Je to z důvodů obsáhlosti seznamu blokovaných virů, který může být poněkud dlouhý, a proto je umístěn na konci filtru. Pojmem skokové pravidlo je na mysli sada více skokových pravidel. Tato pravidla budou tři, protože virus může přijít odkudkoliv, a mířit kamkoliv, je nutné vytvořit pravidlo při vstupu, výstupu a průchodu současně. Je nutno poznamenat, že veškerý provoz bude protékat těmito pravidly. To sebou nese při vysokých průtocích dat i potřebu dostatečného výkonu, aby se nezvětšily latence, či jinak neovlivnila kvalita služeb. A kam bude pravidlo „skákat“? Na konci seznamu budou pravidla, která jednoduše zablokují komunikační porty, na kterých se daný vir šíří, i když je tento způsob jednoduchý, má to obrovskou nevýhodu. Tím, že budou zablokovány porty, je omezena svoboda internetu, a tak se může stát, že i neškodný program, který bude chtít komunikovat na daném portu, má smůlu. V takovém případě je vhodné vytvořit výjimku z pravidla na základě zdrojové IP, která potřebuje danou aplikací využívat. Bez pořádného důvodu není doporučeno výjimky vytvářet.

Skoková pravidla

```
add action=jump chain=input comment="prover virus" disabled=no \
   jump-target=virus
add action=jump chain=forward comment="skoc na virovy filtr" \
    disabled=no jump-target=virus
add action=jump chain=output comment="vyskoc z virusoveho \
    filtru" disabled=no jump-target=virus
```

Vybrané viry

```
add action=drop chain=virus comment="Drop Blaster Worm" \
    disabled=no dst-port=135-139 protocol=tcp
add action=drop chain=virus comment="Drop Messenger Worm" \
    disabled=no dst-port=135-139 protocol=udp
add action=drop chain=virus comment="Drop MyDoom" disabled=no \
    dst-port=1080 protocol=tcp
    add action=drop chain=virus comment="ndm requester" disabled=no\
    dst-port=1363 protocol=tcp
add action=drop chain=virus comment="ndm server" disabled=no \
    dst-port=1364 protocol=tcp
add action=drop chain=virus comment="screen cast" disabled=no \
    dst-port=1368 protocol=tcp
```

Pozornosti neunikne, že specifikace virů nemají ani jednu podmínku prezentovanou na začátku. V tomto případě byl vytvořen vlastní chain s názvem „virus“. Díky tomu skoková pravidla mohou „skákat“ přesně tam, kam je potřeba.

3.7 Point-to-Point Protokol (PPP)

Jedná se o protokol, který umožňuje vzdálené připojení do lokální sítě a umožní být její součástí. Chová se prakticky tak, že uživatel nabývá dojmu, že je skutečně fyzicky připojen do dané LAN sítě (VPN). Využití tohoto protokolu je s určitou nadsázkou nevyhnutelné, a to nejen pro poskytovatele internetu, ale i pro zaměstnance firem, kteří chtějí pracovat z domu, nebo pro jiné účely. V tomto případě zakládáme možnost vzdáleného připojení proto, aby případní technici sítě měli možnost připojení i ze vzdálených pracovišť. Pokud má zařízení v MAN síti veřejnou IP adresu, není jeho správa žádný problém, ale jestliže má adresu privátní, přístup k němu už je problém a bez VPN se k němu nelze dostat. Takováto situace může mít ve chvílích, kdy technik není schopen se do sítě fyzicky připojit a něco opravit, fatální důsledek. Proto je opět doporučeno, ať už kdekoliv na síti, mít alespoň jeden takovýto přístupový bod. MikroTik má toto rozhraní implementováno a práce s ním je jednoduchá, tak jak ji obecně známe z literatury. Pro nastavení VPN potřebujeme v zásadě jen tři věci.

a) Povolení a založení rozhraní PPTP
b) Vytvořit profil, ke kterému budeme přiřazovat účty. Tím bude stanoveno chování účtu
c) Vytváření samotných účtů pro uživatele

3.7.1 Povolení a založení nového rozhraní PPTP

Jako první je povoleno PPTP připojení (založení serveru). V tomto příkazu zadáváme vlastnosti serveru. Můžeme definovat podporu zabezpečení, profil atd.

```
/interface pptp-server server
set authentication=pap,chap,mschap1,mschap2 \
default-profile=default-encryption enabled=yes keepalive-timeout=30\
max-mru=1500 max-mtu=1500 mrru=disabled
```

přidání rozhraní, jediným důležitým parametrem je jeho jméno, a povolení

```
/interface pptp-server
add disabled=no name=pptp-in1 user=""
```

3.7.2 Vytvoření profilu PPTP

Profil definovaný v tomto příkazu se bude jmenovat „one“ a jak už jméno napovídá, bude sloužit pro připojení pouze jediného uživatele. Tento uživatel dostane pokaždé stejnou veřejnou adresu z přiděleného rozsahu, a to 213.192.39.201. Dále je definováno, že kódovaní je povoleno, ale bez komprese. Kdyby bylo potřeba připojit na jeden profil více uživatelů, bylo by nutné změnit parametr „only-one=no“ a do parametru „remote-address“ přiřadit nějaký pool. Pool určuje rozsah IP adres a jeden pool může být přiřazen více funkcím, které tento rozsah používají. Sdílení poolu není však vhodné z důvodů přehlednosti a kolizím čerpaných adres. Ještě je nutno dodat, že tento profil je obecný pro všechna tunelovaná připojení (L2TP, atd.).

```
/ppp profile
add change-tcp-mss=yes remote-address=213.192.39.201 \
    dns-server=213.192.40.6 local-address=213.192.39.200 name=one\
    only-one=yes remote-ipv6-prefix-pool=none use-compression=no \
    use-encryption=yes use-ipv6=yes use-mpls=default \
    use-vj-compression=no
```

3.7.3 Vytvoření účtu PPTP

Účet, který bude vytvořen, bude používat přihlašovací jméno admin, jeho heslo bude také admin, bude vázan na profil one a přístup bude mít jen ke službě PPTP.

```
/ppp secret
add caller-id="" disabled=no limit-bytes-in=0 limit-bytes-out=0 \
    name=admin password=admin profile=one routes="" service=PPTP
```

Kdyby bylo potřeba, aby účet požíval všechny dostupné metody tunelového připojení, musí být zadána hodnota parametru „service=any“ [6]

4 Gigabitový směrovač v roli retranslační stanice

Jedná se o takovou stanici, která přijímá službu a dále ji distribuuje. Distribuovat může buďto dalším r. stanicím, koncovým stanicím, nebo obojí. Pojmem služba je myšleno v tomto případě telekomunikační datový okruh, který je šířen takřka jakýmkoliv fyzickým médiem. (802.11a/b/g/n, 10GHz, 24GHz, 80GHz, optika, Ethernet, atd.).

Tato kapitola bude zaměřena na kombinovanou retranslační stanici. Taková stanice totiž zaštituje potřeby aplikovat filtry pro komunikaci a přístup do sítě, priority komunikace, přístupové metody koncových stanic, přidělování šířky pásma koncovým stanicím a další. Jsou to právě tato nastavení, která definují chování sítě a tvoří tak celé know-how ISP. Některé z potřebných metod pro implementaci RouterOS jako retranslační stanici, zde budou demonstrovány. Obecně se bude pracovat s architekturou x86, jejíž základ bude tvořit základní deska SuperMicro PDSMA+, která bude osazená čtyřjádrovým procesorem Intel Xeon 2,3GHz a 2GB RAM. Topologie je totožná s uvedenou topologií v příloze, na niž odkazuje kapitola 3. Konkrétně se bude jednat o Retrans 2.

4.1 Potřebná rozhraní a jejich účel

Retranslační stanice bude mít celkem pět rozhraní. Dvě rozhraní RJ45, pracující v módu full duplex gigabit, a tři VLAN rozhraní. Rozhraní eth2 na desce bude sloužit jako trunk port do L2 přepínače, který je také Full gigabit. Na tento přepínač jsou připojeny dvě směrové antény s kapacitou 500MB a jedna sektorová anténa sloužící k distribuci datových služeb koncovým stanicím. Eth1 na desce bude sloužit pro příjem služby ze serveru NAT. Příjem je zajištěn v pásmu 10GHz.

Nejprve budou tedy založeny potřebné VLAN-y. Postup bude totožný jako v předchozí kapitole, jen hodnoty VLAN ID budou rozdílné.

```
/Interface vlan
add arp=enabled disabled=no interface=ether2trunk mtu=1500 \
    name=vlan11 use-service-tag=no vlan-id=11
add arp=enabled disabled=no interface=ether2trunk mtu=1500 \
    name=vlan12 use-service-tag=no vlan-id=12
add arp=enabled disabled=no interface=ether2trunk mtu=1500 \
    name=vlan12 use-service-tag=no vlan-id=13
```

4.2 Přiřazení IP adres

Pro následující ukázku bude potřeba následujících adres:

a) Transportní ze serveru NAT
b) Transportní na Retrans 4 a 5
c) Transportní rozsahy pro koncové stanice (založíme 4).

/ip address

```
add address=93.99.109.50/30 comment="transport NAT" \
    disabled=no interface=eth1 network=93.99.109.48
add address=93.99.109.65/30 comment="transport retrans4" \
    disabled=no interface=vlan12 network=93.99.109.64
add address=93.99.109.69/30 comment="transport retrans5" \
    disabled=no interface=vlan13 network=93.99.109.68
```

a pár klientských rozsahů

```
add address=172.20.90.1/29 comment="klient 1" \
    disabled=no interface=vlan11 network=172.20.90.0

add address=172.20.90.9/29 comment="klient 2" \
    disabled=no interface=vlan11 network=172.20.90.8

add address=172.20.90.17 comment="klient 3" \
    disabled=no interface=vlan11 network=172.20.90.16

add address=172.20.90.25 comment="klient 4" \
    disabled=no interface=vlan11 network=172.20.90.24
```

Tímto bylo definováno komunikační rozhraní a IP adresy. Následně budou sestavovány služby nad těmito rozhraními, které bude retranslační stanice používat.

4.3 Využití pravidel filtru a překladu adres na retranslační stanici

Pro implementaci zabezpečení a aplikaci filtračních pravidel na retranslační stanici bude potřeba aplikovat jiná pravidla než na serveru NAT. Zdá se to celkem logické, protože v každé ukázce výše plní RouterOS jinou roli, proto je tedy nutné k tomu přizpůsobit i pravidla. Aplikovaná pravidla jsou jiná také z hlediska duplicity. Příklad takového pravidla je třeba hlídaní SSH útoku. Jsou–li tyto útoky hlídané na severu NAT, není je potřeba znovu zadávat do retranslačních stanic. Nejvíce se však liší v zásadě v tom, že na retranslační stanici jsou připojené koncové stanice. Pro ty musíme stanovit pravidla, aby na naší síti nebyl chaos.

Jedná se hlavně o tato pravidla.

a) Povol potřebné adresy z rozsahu, zbytek zakaž
b) Zakaž p2p na koncových stanicích
c) Povol koncové stanici pouze vybrané servery

4.3.1 Přístup do sítě pouze vybraným IP adresám

Pro udržení pořádku a přehledu na síti je možné definovat, které IP adresy mohu do sítě přistupovat. Je to dáno tím, že žádný poskytovatel nechce, aby na síti existovaly adresy, které není schopen identifikovat. Neschopnost identifikace IP adresy může způsobit, že poskytovatel nebude schopen jednoznačně říci, zda daná IP adresa má vůbec právo do sítě přistupovat, nebo v případně potřeby, pro ni aplikovat nějaká individuální pravidla. V následující modelové situaci bude dáván koncovým stanicím rozsah osmi adres (šest použitelných). Následně je předpokládáno, že pro komunikaci potřebuje koncová stanice pouze jednu adresu (ostatní jsou jako reserva). V tu chvíli je nežádoucí, aby ostatní volné adresy byly „aktivní“, ba naopak je žádoucí, aby byly pozastaveny až do doby jejich využití.

Pravidla se dělí na dvě části. V té první bude povolena vybrána adresa v obou směrech, kdežto v druhé části bude zakázán zbytek rozsahu v obou směrech.

První část, povol jednu IP v obou směrech.

```
/ip firewall filter
add action=accept chain=forward comment=koncova_stanice_1 \
    disabled=no src-address=172.20.90.3
add action=accept chain=forward comment= koncova_stanice_1 \
    disabled=no dst-address=172.20.90.3
```

Druhá část, zakaž zbytek v obou směrech

```
add action=drop chain=forward comment=koncova_stanice_1_drop \
    disabled=no src-address=172.20.90.0/29

add action=drop chain=forward comment=koncova_stanice_1_drop \
    disabled=no dst-address=172.20.90.0/29
```

Touto definicí bude tedy moci z celého rozsahu 172.20.160.16/29 komunikovat pouze jedná adresa. Vybranou adresou je 172.20.160.19. [1,4]

4.3.1 Omezení provozu P2P komunikace

Proč omezovat komunikaci na P2P portech, a tím opět narušovat „svobodu internetu“? Z pohledu uživatelů, kteří P2P komunikaci využívají, je tento postoj ISP naprosto nepochopitelný a politováníhodný. Z pohledu ISP je však tento postoj vnímán daleko jinak. Provoz na P2P je nežádoucí z toho důvodu, protože generuje velký počet malých paketů, které komunikaci na bezdrátových rozhraních normy 802.11a/b/g/n zásadním způsobem ovlivňují. Je to způsobeno tím, že norma 802.11 a/b/g/n přijímá i vysílá v jednom kanále (norma 802.11n přes dva kanály). Rádiová karta se tedy pokaždé musí přepnout z vysílání na příjem a naopak. Při velkém počtu paketů za sekundu a větším počtu uživatelů P2P je tento Access point doslova „zabíjen“ neustálým přepínáním Rx do Tx a zpět. Tento faktor má pak vliv na latence, rychlost, a tím na celou kvalitu služeb. Navíc se dá se říci, že na P2P sítích je nejčastěji šířen nelegální obsah dat tzv. warez přes torrenty. Výsledkem zablokování této komunikace je tedy nižší HW vytížení, zlepšení kvality služeb a vyvarování se nepříjemným problémům s nelegální distribucí dat. K celému omezení provozu P2P postačí dvě jednoduchá pravidla, neboť MiktoTik sám nabízí kvalitní možnosti omezení tohoto provozu. Omezení bude probíhat opět v obou směrech.

```
/ip firewall filter
add action=drop chain=forward comment=drop_p2p \
    disabled=no dst-address=172.20.0.0/16 p2p=all-p2p
add action=drop chain=forward comment=drop_p2p \
    disabled=no p2p=all-p2p src-address=172.20.0.0/16
```

Je možné si povšimnout, že je opravdu provoz zakázán každému s klientským rozsahem 172.20.x.y.

4.3.2 Povolení komunikace pouze s vybranými servery

Další z věcí, která je běžně v praxi nasazována, je povolení komunikace pouze s vybranými servery, nebo povolení komunikace jen na určitých portech. Taková situace může nastat třeba v případě, kdy má dané zařízení pouze jeden účel. Může se jednat o kamery, které monitorují nějaké místo, a jejich stream se přenáší na konkrétního DVR zařízení. Další modelovou situací může být umístění PC v nějaké restauraci, hotelu, které slouží pouze pro rezervace a musí mít přístup k databázovému serveru. Samozřejmě situací, kdy se hodí uplatnit toto omezení, může být víc. Následující ukázka bude zaměřena na to, jakým způsobem lze vytvořit seznam vybraných serverů, na které bude moci uživatel přistupovat. Zavedení této restrikce se bude opět skládat z několika pravidel. V prvních řádcích budou uvedeny servery, na které bude komunikace povolena. V dalším pravidle bude zase uvedeno, kdo se má omezit a jaký port tedy povolit. Poslední pravidlo zakáže vše ostatní kromě výše povoleného.

Servery, na které je přístup povolen.

```
/ip firewall nat
add action=accept chain=dstnat comment=enable_seznam.cz disabled=no\
    dst-address=77.75.76.3
add action=accept chain=dstnat comment=enable_2 disabled=no\
    dst-address=2.2.2.2
add action=accept chain=dstnat comment=enable_server3 disabled=no\
    dst-address=3.3.3.3
```

Je vidět, že povolení serveru není nic jiného než jen akceptování překladu cíle.

Nyní pravidla určující, kdo má být omezen a jakým způsobem.

```
add action=dst-nat chain=dstnat comment=disabled_user \
    disabled=no dst-port=!80 protocol=tcp src-address=172.20.92.171\
    to-addresses=172.20.92.171
```

V pravidle je řečeno, že cokoliv kromě portu 80 má být přesměrováno zpět na zdrojovou adresu. Díky tomu, že bylo v předcházejících pravidlech už překládání konkrétních serverů povoleno, priorita říká, že na tyto servery se pravidlo vztahovat nebude. Kdyby byla zapotřebí webová stránka, kde budou vypsány všechny povolené servery a uživatel jen klikne na jeden z nich, bude pravidlo vypadat takto.

```
add action=dst-nat chain=dstnat comment=disabled_user \
    disabled=no dst-port=80 protocol=tcp src-address=172.20.92.171\
    to-addresses=4.4.4.4 to-ports=80
```

Místo adresy 4.4.4, která je jenom ilustrativní, bude zadána vlastní adresa. To samé platí i o parametru „to-port="

4.4 Přidělování šířky pásma, inteligentní řízení provozu

Další nevyhnutelnou a naprosto běžnou součástí sítí je přidělování šířky pásma, jinak řečeno rychlosti. MikroTik nabízí velice propracované možnosti, jak toho docílit. Od nejjednodušších omezení, která se dají nastavit na samotných rozhraních, nebo každé bezdrátové koncové stanici zvlášť, která je v Access Listu, přes centrální Simple Queues až po inteligentní Queue tree.

Před samotnou praktickou ukázkou si je třeba vysvětlit základní pojmy, se kterými bude pracováno.

Max. limit – Maximální rychlost ve tvaru uplod/download

Burst limit – Maximální rychlost v burstu

Burst threshold – Maximální limit rychlosti po uplynutí Burst time

Burst time – Doba, po kterou bude uplatněn Burst limit (maximální rychlost)

Limit at – Generovaná rychlost ve tvaru uploď/download, od které se bude odpočítávat Burst time.

V praktických demonstracích, zde budou uvedeny způsoby nastavení Simple Queues a Queue tree.

4.4.1 Simle Queues

Název tohoto způsobu řízení přiděleného pásma je velice trefný. Nastavení v tomto případě je opravdu jednoduché a používá se v tu chvíli, kdy je potřeba omezení aplikovat na konkrétní adresy či subnety. V tomto příkladu bude omezen klient_1 následujícím způsobem. Maximální rychlost bude 8M po dobu 5 minut, pak klesne na 4M. Odpočítávání bude probíhat od rychlosti 6M.

```
add burst-limit=8M/8M burst-threshold=4M/4M burst-time=5m/5m \
    direction=both disabled=no interface=all limit-at=0/0 \
    max-limit=6M/6M name=queue4 parent=none \
    priority=8 queue=default-small/default-small \
    target-addresses=172.20.90.1/32 total-queue=default-small
```

4.4.2 Queue tree

Zatím co u Simple Queses bylo možné použít pouze pro omezení konkrétní IP adresy, nebo rozsahu, u Queue tree jsou možnosti použití daleko širší. Queue tree je uplatněno v těch situacích, kdy je potřeba řídit kvalitu služeb na síti (QoS), nebo inteligentně přidělovat volné pásmo, které se na lince nachází. Modelový scénář, v jakém případě je možné použití tento sofistifikovaný systém, je následující. Je dána sektorová anténa, na které se nachází několik klientů. Určitým způsobem (testy, výpočty) bylo spočítáno, že dané rozhraní, na které je tato sektorová anténa připojena, dokáže distribuovat bezpečně 20MB. Dá se předpokládat, že klienti nebudou plně využívat kapacity své linky současně. V tom případě je k dispozici nadbytek kapacity, kterou je možné dočasně využít. Využití bude takové, že to co je „navíc“, bude přiděleno nad rámec tarifu koncovým klientům, kteří v danou chvíli generují provoz na lince na tomto rozhraní. Toto má za důsledek zvýšenou spokojenost klientů, což je určitě vyhovující. Dojde-li k tomu, že klient, který nevyužíval své kapacity, nebo vůbec negeneroval provoz na lince, začne linku využívat, systém automaticky odebere pásmo, které přidělil nad rámec ostatním, a přidělí jej klientovi, který linku nevyužíval. Tímto způsobem se dá říci, že je možné prodat více konektivity, než se ve skutečnosti na daném rozhraní nachází. Zní to pěkně, ale i to má své úskalí. Existuje parametr CIR, který je v mnohých státech definován i zákonem. Jedná se o poměr prodané konektivity k dostupné a nesmí přesáhnout určitou mez. Tato hodnota je desetinné číslo př. 0,8. Samozřejmě nejideálnějším případem je, že CIR=1, tj. dostupná konektivita je rovná prodané. Celý výše zmíněný model pracuje se stanicemi na jednom sektoru, avšak toto celé se dá uplatnit i nad rozhraním, které slouží jako hlavní konektivita r. stanice. To znamená, že zmíněné rozhraní, na které je připojena směrová anténa, je celé podřízené příjmovému rozhraní. V praxi to pak vypadá tak, že na příjmu se může nacházet třeba 50MB, které se dělí mezi rozhraní s klienty a dvěma rozhraními, které směřují k dalším r. stanicím. Každé toto rozhraní má definovanou svou maximální propustnost a to včetně burst limitu. Pod těmito rozhraními jsou pak závislé klientské linky, které se dělí na dvě pravidla uploadu a downloadu. Tvoření těchto závislostí pak vytváří strom definic omezení, proto název Queue tree.

V ukázce bude předvedeno, jak takové inteligentní řízení provozu nastavit pro čtyři klienty. Nastavení se bude vztahovat k výše definovaným klientům „klientská_stanice_1 - 4“

4.4.2.1 Nastavení Mangle

Queue tree nepracuje samo o sobě, nebo s IP adresami stanic. Ke své práci používá označené pakety, které musíme označit v mangle. Každá stanice musí mít definována dvě pravidla pro označování. Jedno pravidlo bude označovat pakety downloadu a druhé pakety uploadu. Je nutné dodat, že pravidla pro označování paketu se chovají jinak než pravidla z firewallu. Je-li hodnota Action= accept, pravidlo se provede, ale už se nezkoumají další. Je-li passthrough, pravidlo se aplikuje, ale

paket se testuje i na další atd. V tomto případě bude použitá akce „mark packet“. Celé pravidlo pak zní. Je-li paket před směrováním ze zdrojové IP adresy a.b.c.d, pak ho označ jako upload. A samozřejmě opačné pravidlo. Je-li paket před směrováním pro cílovou adresu a.b.c.d, pak ho označ jako download. Tato dvě pravidla vytvoříme pro každou klientskou stanici zvlášť.

```
/ip firewall mangle
add action=mark-packet chain=prerouting comment=stanice_1_up \
    disabled=no new-packet-mark=stanice_1_up passthrough=no \
    src-address=172.20.90.3
add action=mark-packet chain=prerouting comment=stanice_1_down \
    disabled=no new-packet-mark=stanice_1_down passthrough=no \
    dst-address=172.20.90.3
add action=mark-packet chain=prerouting comment=stanice_2_up \
    disabled=no new-packet-mark=stanice_2_up passthrough=no \
    src-address=172.20.90.11
add action=mark-packet chain=prerouting comment=stanice_2_down \
    disabled=no new-packet-mark=stanice_2_down passthrough=no \
    dst-address=172.20.90.11
```

Teď již jsou pakety označeny. Samozřejmostí je, že pro zbývající stanice se bude postupovat analogicky.

4.4.2.1 Vytváření Queue tree

Teď, když je veškerý potřebný provoz označen, je možné sestavovat strom závislostí. Nejprve bude založené pravidlo, kde bude nastavena celková propustnost příjmu r. stanice. Toto pravidlo nemá žádného nadřazeného rodiče, proto je použito virtuální pravidlo globálního příjmu „global-in“.

```
/queue tree
add burst-limit=50M burst-threshold=30M burst-time=4m disabled=no\
    limit-at=35M max-limit=30M name=prijem packet-mark=""  \
    parent=global-in priority=2
```

Tímto je definováno, že queue bude mít maximální rychlost v burstu 50MB po dobu 4 minut. Po této době klesne na 30MB a odpočítávat se bude při provozu nad 35MB. Tyto parametry jsou na bezpečné hranici propustnosti rozhraní tvořeného normou 802.11a v módu Nstreme. Další definicí bude pravidlo pro rozhraní na sektorové anténě, respektive „klientské“ anténě.

```
add burst-limit=15M burst-threshold=10M burst-time=2m disabled=no\
    limit-at=9M max-limit=10M name=sektor_klienti packet-mark="" \
    parent=prijem priority=3 queue=default
```

Je viditelné, že obě pravidla mají jinou prioritu. Podřazené má menší číslo než nadřazené a závislost je „parent=prijem“. Následující pravidlo je už globální definice pro provozu klienta.

```
add burst-limit=5M burst-threshold=2M burst-time=2m disabled=no
    limit-at=1500k \
    max-limit=2M name=stanice_1_omezeni packet-mark=stanice_1 \
    parent=sektor_klienti priority=7
```

Na konci budou na předcházejícím pravidle závislá následující pravidla pro upload a download.

```
add burst-limit=5M burst-threshold=2M burst-time=2m disabled=no \
    limit-at=0 max-limit=0 name=stanice_1_down queue=default \
    packet-mark=stanice_1_down parent=stanice_1_omezeni priority=7
add burst-limit=0 burst-threshold=0 burst-time=0s disabled=no \
    limit-at=0 max-limit=0 name=stanice_1_up queue=default  \
    packet-mark=stanice_1_up parent=stanice_1_omezeni priority=5
```

Těmito pravidly byly definovány závislosti na řízení provozu. Vlastnosti těchto pravidel jsou následující. Jedná se o half duplexní linku, protože v nadřazeném pravidle „stanice_1_ omezení“ jsou definována maximální rychlostí celé linky, bude-li download na 3MB/s z 5, pro upload zůstanou 2MB/s a naopak. Proto je možné zpozorovat, že v pravidle „stanice_1_up“ není definovaná žádná hodnota, ta je odvozená z aktuálního provozu na downloadu. Klient ve výsledku toto moc nepozná, bude-li měřit rychlost, vždy se měří nejprve download a pak upload, případně naopak, ale nikdy ne společně. Kdyby ale měřil jen např. upload a zároveň by něco stahoval, bude hodnota menší. Kdyby toto bylo nežádoucí, nepsalo by se pravidlo nadřazené „stanice_1_omezeni“, ale bylo by napsáno rovnou pravidla pro up a down. V tomto případě by pravidlo pro upload muselo mít nastavené hodnoty. A byly by závislé na rodiči „sektor_klienti“. Taková linka by byla pak full duplex.

[6]

4.5 Způsoby připojení koncových stanic k retranslační stanici, ACL

O tom, jakým způsobem připojovat koncové stanice k r. stanici se dá vést diskuze. Každá síť preferuje jiné mechanismy a s těmi pracuje. Příkladové možnosti mohou být

a) Statické připojení
b) Dynamické
c) PPoE
d) HotSpot

Každé z těchto druhů připojení má své výhody i nevýhody. Správce sítě by se měl rozhodnout na základně toho, jaké služby plánuje současně s těmi datovými (internet) nabízet. Rozhodne-li se například provozovat na své síti IP TV, rozhodně nepoužije HotSpot. Pokud však chce mít klientské rozhraní z jakéhokoliv důvodu veřejné, naopak HotSpot použije jako nejlepší variantu. [3]

4.5.1 Statické připojení koncových stanic

Pojmenování tohoto druhu připojení není oficiálním názvem, ale dá se tak nazvat, protože se jedna o připojení, které má napevno a napořád definovanou IP adresu. Představa takovéhoto připojení může být následující. Stanice je připojena fyzickým médiem (802.11, RJ45, SFP) a má přidělenou statickou adresu. Tato adresa může být definována buď přímo v PC, nebo do směrovače (modemu) na rozhraní WAN. Směrovač pak na rozhraní LAN rozděluje pomocí DHCP adresy lokálního rozsahu 192.168.x.y/24. V tomto případě pak musí směrovač zároveň podporovat funkci NAT, aby se celá klientská síť mohla překládat na adresu poskytovatele. Tato možnost je hojně preferována díky tomu, že postačí pouze jeden transportní rozsah /30 a přitom klient může mít doma takřka neomezený počet zařízení. Zároveň je na retranslační stanici prováděno jen směrování, takže nároky na výpočetní výkon ke zpracování paketu jsou minimální. Zapojení přímo do PC není doporučeno. I když se může ISP snažit aplikovat na síti co nejlepší a co nejúčinnější pravidla proti nežádoucím vlivům, nemůže být to 100% zajištěno. V kombinaci se špatně nastaveným firewallem v PC je hrozba infikace virem vysoká, nehledě na to, že nastavením firewallu u osobních PC často běžný uživatel ani nevěnuje pozornost, ba dokonce ani o existenci této možnosti netuší. Tento způsob byl použit i v ukázkách, kde byly tvořeny sítě s maskou /29.

4.5.2 Dynamické připojení koncových stanic

Jedná se prakticky o totožný způsob připojení. Rozdíl je akorát v tom, že adresy jsou přiřazovány koncovým stanicím serverem DHCP, který je spuštěn na retranslační stanici. Tuto možnost připojení lze doporučit výhradně u domácích, či firemních, sítí. Nevýhodu pro ISP má totiž tu, že není jednoznačně daná adresa stanice, a tak případná nastavená pravidla pro její omezení, nebo řízení, mohou selhat. Dá se sice stanovit konkrétní MAC adresa zařízení, na kterou bude přidělována stejná IP adresa, ale takové řešení je už opravdu totožné s předcházejícím a akorát více stoupne

zatížení r. stanici tím, že se musí starat o seznam přidělených adres a o provoz DHCP. Dá se říci, že tyto dvě funkce dohromady zas až tolik výpočetního výkonu nevezmou, což je pravda. Naopak se ale dá oponovat, čím více funkcí je použito, tím je větší pravděpodobnost, že může nějaká selhat a tím zbytečně omezit stanici. Obě předchozí varianty mají společnou nevýhodu. Pro připojení jedné stanice je za potřebí minimálně 4 adresy. Network IP, GW IP, station IP, Broadcast.

4.5.3 Připojení pomocí protokolu PPoE

Tento typ připojení je druhým nečastějším typem připojení k r. stanici, které se u bezdrátových ISP vyskytuje. U poskytovatelů kabelových připojení xDSL je tato metoda běžným standardem. Tato technologie je ideálním řešením do sítí, které mají společné přístupové médium, ale je žádoucí, aby se linka chovala jako point-to-point. Už z popisu je jasné, že tyto dva požadavky jsou si protichůdné, ale v praxi často požadované. PPPoE navíc neřeší problém jen se sdíleným médiem, ale také problém s ověřováním přístupu. Pro připojení je potřeba totiž uživatelské jméno a heslo, se kterým se klient vůči retranslační stanici bude autorizovat. Tato vlastnost je pro ISP v jistém měřítku dobrá. V předchozím statickém připojení bylo nutné zřídit autorizaci a zabezpečit vysílání. Autorizace je aplikována buď jednou z šifrovacích metod, jako jsou WEP/WPA/WPA2/atd., nebo ACL pomocí MAC adres zadaných do Access listu. Bylo to z toho důvodu, aby nemohla žádná třetí strana získat fyzický přístup a tím odposlouchávat data. V případě PPPoE zadáme do koncové stanice zmíněné jméno a heslo a tím je autorizace jednoznačná. Má to ale chybu, PPPoE řeší přístup pouze k TCP/IP vůči „přístupovému bodu“, ale ne na fyzické médium. Takže pro opravdu zabezpečené připojení je víc než vhodné na bezdrátových sítích používat jednu z výše uvedených metod zabezpečení současně s PPPoE.

Následující příklad bude věnován nastavení přístupového bodu PPPoE, který bude autorizovat koncové stanice domovního kabelového rozvodu. Tento rozvod je realizován přepínanou sítí. Server bude mít k dispozici rozsah 172.16.1.0 /26 čili 64 adres. Když se odečtou adresy pro síť, oběžník a GW, je k dispozici 61 adres, které je možné přiřazovat výše zmiňovaným klientům. Nastavení PPPoE není nikterak těžké. Nejprve je založen nový rozsah na rozhraní, kde bude PPPoE provozováno, a poté je založen nový pool, který bude k dispozici pro PPPoE server. Pak bude nastaven profil, který bude využívat nastavený pool a bude definovat možnosti autorizace s protokoly přenosu. Jako poslední bude nastaven samotný PPPoE server se založením jednoho uživatele.

Přidání rozsahu na rozhraní

/ip address

```
add address=172.16.1.1/26 comment=PPPoE disabled=no interface=ether2
network=172.16.1.0
```

Založení nového poolu

```
/ip pool
add name=pool_PPPoE ranges=172.16.1.2-172.16.1.62
```

Profil, který bude aplikován na uživatele PPPoE

/ppp profile

```
add change-tcp-mss=no dns-server=77.48.100.254 \
    local-address=172.16.1.1 name=profile_PPPoE only-one=default \
    remote-address=pool_PPPoE use-compression=no \
    use-encryption=yes use-ipv6=no use-mpls=no use-vj-compression=no
```

Založení PPPoE serveru a uživatele

```
/interface pppoe-server server
add authentication=pap,chap,mschap1,mschap2
    default-profile=profile_PPPoE disabled=no interface=ether1 \
    keepalive-timeout=10 max-mru=1480 max-mtu=1480 \
    max-sessions=0 mrru=disabled one-session-per-host=no \
    service-name=PPPoE_klienti
/ppp secret
add caller-id=1 disabled=no limit-bytes-in=0 limit-bytes-out=0
    name=pppklient password=55555 profile=default routes=""
    service=pppoe
```

5 Dynamické směrování

Velikost dnešních IP sítí je tak rozsáhlá, že je nad rámec lidských schopností tvořit a spravovat směrovací tabulku ručně. V takovém případě je téměř stoprocentní pravděpodobnost zanešení chyby. Najít při tom takovou chybu je někdy hodně obtížné. Bude-li navíc do MAN sítě integrována záložní konektivita, nebo více retranslačních stanic s redundantním příjem konektivity, je nemožné bezpečně „otáčet“ provoz podle výpadku na síti. Když nějaká mezilehlá r. stanice selže, všechny zazní také. V lehčím selhání je možnost pomocí MAC telnetu tuto chybu opravit vzdáleně. Jestli však r. stanice selže fyzicky, je nutná návštěva technika. V takovém případě jsou všechny r. stanice ležící za tou poškozenou ve stavu „down“, protože nebyl mechanismus, co by otočil provoz. Absolutně opačná situace nastává, když je na síti aplikována jedna z variant autonomního směrování, které nám MikroTik nabízí. V zásadě nejběžnějšími protokoly, které jsou pro IPv4 síť použitelné, jsou RIP, OSPF, BGP. Jakýkoliv z těchto protokolů je doporučeno použít, i kdyby MAN síť nedosahovala velkých rozměrů, neboť i na malé síti se dá udělat při manuálním směrování dost chyb. Navíc je pohodlnější, když pro každého nového klienta není potřeba zadávat i cestu skrze síť. Naprosto nešťastné řešení je přidělení každému směrovači jen default GW ve stylu 0.0.0.0/0 -> GW_IP. Jedním z důvodů je to, že tímto nastavením není možné se pohybovat ani mezi jednotlivými koncovými stanicemi v MAN síti. Pro odstranění takovýchto problémů je vhodné alespoň použít protokol RIP. Nasazení tohoto protokolu je velmi jednoduché a splní účel. Sice neumí otáčet provoz, ale sofistikovaně řeší distribuci rout po síti. Pro správce, kteří chtějí něco víc, jsou tu protokoly OSPF a BGP. Oba dokáží otáčet provoz (měnit default gw na základě dostupnosti). Oba samozřejmě umí i distribuci rout. Rozdíl je však v tom, jak rychle dokáží propagovat změny v síti. Při pokusné implementaci obou protokolů na různých, ale podobných částech sítě bylo zjištěno pomocí úmyslných výpadků, že BGP reaguje na změnu až o 60% rychleji než OSPF. Po implementaci byl uskutečněn hovor pomocí VoIP. Během hovoru byl provoz sítě přesměrován. VoIP spojení jakožto spojení, které patří mezi ty, které jsou náchylné na latence a změny během hovoru, téměř nic nepoznalo. Jediné, co se projevilo, bylo „zaškobrtnutí hlasu“, hovor však dále plynule pokračoval. Je tedy na správci a jeho schopnostech, jaký protokol využije pro své řešení. K této tématice bude prezentováno základní nastavení BGP protokolu. Vhodnou poznámkou je, že tento protokol je aplikován i na mnoho WAN sítích (národních, mezinárodních), takže pokud ISP zažádá o vlastní ASN číslo, není problém tento protokol využít a být přímo součástí RIPE jako LIR. Pro tento krok je však už opravdu nezbytné, aby správce tento protokol ovládal, neboť špatná nastavení (třeba ASN) může způsobit kolaps páteřní sítě v národním měřítku. Důsledek může být hodně nepříjemný.[1,4]

5.1 Základní implementace BGP

Následující testovací topologie bude triviální. Bude se skládat ze dvou retranslačních stanic, které budou propojeny jedním peerem. BGP má v základu dvě věci, které musí být nastaveny, aby byla komunikace funkční. Je to interface a peer. Interface definuje, co se bude do sítě pomocí peeru přenášet. Může například redistribuovat směrovací tabulky vytvořené jinými protokoly, nebo aplikovat různé filtry. Nejdůležitější je však údaj o AS číslu, který je potřebný pro celou definici BGP a nesmí se opakovat jinde v síti. Po vytvoření interface je vytvořen peer. V této položce je definováno vůbec celé chování distribuce dat protokolu BGP. V nastavení distribuce je například zakomponovaná možnost, jestli se má propagovat defaultní routa, nebo jestli má být propagováno i IPv6. Dále je zde definována IP adresa vzdáleného bodu, ke kterému je peer připojen, atd.

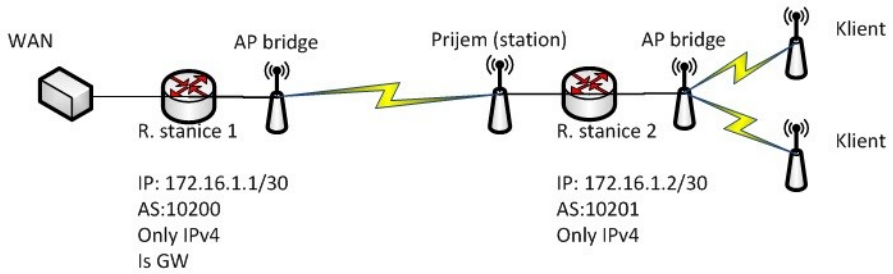
Parametry retranslačních stanic:

R . stanice 1

-IP: 172.16.1.1/30
-AS:10200
-Distribuce pouze IPv4
-Nedistribuuje žádné jiné směrovací protokoly.
-Je default GW pro r. stanici 2

R. stanice 2

-IP: 172.16.1.2/30
-AS: 10201
-Přijímá pouze směrování IPv4
-Nedistribuuje žádnou def. GW



Obrázek 5.1 schéma zapojení pro test BGP

5.1.1 Nastavení R. stanice 1

Prvním krokem je nastavení interface

```
/routing bgp instance
set default as=10200 client-to-client-reflection=yes disabled=no \
    ignore-as-path-len=no name=default out-filter="" \
    redistribute-connected=yes redistribute-ospf=no \
    redistribute-other-bgp=yes redistribute-rip=no \
    redistribute-static=yes router-id=0.0.0.0 routing-table=""
```

Po nastavení interface je nastaven peer

```
add address-families=ip as-override=no default-originate=always
    disabled=no hold-time=3m in-filter="" instance=default \
    multihop=no name=peerStanice2 nexthop-choice=default \
    out-filter="" passive=no remote-address=172.16.1.2 \
    remote-as=10201 remove-private-as=no route-reflect=no \
    tcp-md5-key="" ttl=default use-bfd=no
```

5.1.2 Nastavení R. stanice 2

```
/routing bgp instance
set default as=10201 client-to-client-reflection=yes disabled=no \
    ignore-as-path-len=no name=default out-filter="" \
    redistribute-connected=yes redistribute-ospf=no \
    redistribute-other-bgp=yes redistribute-rip=no \
    redistribute-static=yes router-id=0.0.0.0 routing-table=""
```

a opět peer

```
add address-families=ip as-override=no default-originate=never
    disabled=no hold-time=3m in-filter="" instance=default \
    multihop=no name=peerStanice2 nexthop-choice=default \
    out-filter="" passive=no remote-address=172.16.1.1 \
    remote-as=10200 remove-private-as=no route-reflect=no \
    tcp-md5-key="" ttl=default use-bfd=no
```

Těmito dvěma nastaveními na každém směrovači bylo docíleno, že peer naváže spojení (status=estabilished). V /ip route je pak možno vidět, že směrovací tabulka obsahuje informace o nových cestách z druhé stanice. Pro definování propagace defaultní GW se parametr „default-originate“, který je na retranslační stanici 1 nastaven jako „always“, tedy bude propagovat vždy. Na retranslační stanici 2 je nastaven jako „never“, tudíž nebude nikdy propagovat, protože je to koncová stanice v síti. [6]

6 IPv6

Již dlouhou dobu je znám obecný fakt, že rozsah adres z protokolu IPv4 je téměř vyčerpán. Avšak již dlouhou dobu je také vyvíjen nástupce tohoto protokolu jménem IPv6. Počátek vývoje se datuje už v první polovině devadesátých let, ale vývoj byl velice pomalý, protože výrobci HW a technologie pro sítě viděli business v aktuálním protokolu IPv4, a proto nechtěli investovat do podpory IPv6. Navíc se nacházely dodatečné techniky, které měly rychle zmenšující se rozsah IPv4 zpomalit. Takovou technologií je notorický známý NAT. Samotní uživatelé nic nenamítali, protože v devadesátých letech měl internet opravdu málokdo. Představa, že je možné s datovým připojením nakládat i jinak než jen na prohlížení webových stránek, byla zcestná, protože nebyla ani domácí elektronika, která by toto připojení využila. Z těchto důvodů nebyla ani potřeba přidělovat uživatelům veřejnou IP adresu. Toto mělo za následek, že se NAT s velkou oblibou rozšířil. Dnešní situace je však opačná, existuje spoustu domácí elektroniky, která připojení k internetu umožňuje a nabízí skrze něj služby. Mluvíme třeba o domácích NAS uložištích, které chceme mít dostupné i mimo domov. Také spousta aplikací, jako je VoIP, nemají pro svou činnost moc rádi NAT, lepší je přímé spojení. A jako poslední fakt je to, že internet měl od počátku myšlenku, že uživatelé mají možnost mezi sebou komunikovat nezávisle a přímo. Toto vše vede k většímu počtu žádostí o veřejnou IP adresu, ale IPv4 je nedokáže vyslyšet. Zde se nachází zásadní průlom. Kde je poptávka, tvoří se nabídka. A tak se setkáváme čím dál více s technologií, která IPv6 podporuje. Jsou také daleko rychleji prakticky aplikovány standardy, které byly zatím jen na papíře v podobě RFC. Jediné, co je potřeba ještě rozšířit, je prezentace IPv6 správcům a dát jim důvod, proč IPv6 začít nasazovat. V současné době je možné se setkat jen s pár správci, kteří mají zájem a chuť IPv6 nasazovat. Zatím je trend označovat IPv6 za složitý protokol, který zatím není důvod nasazovat. To je však omyl. Včasná implementace umožní mít konkurenční náskok před dalšími ISP, a zvýší se portfolio služeb. V následující demonstraci nasazení IPv6 je předpokládána jeho částečná znalost.

Jako první krok pro nasazení protokolu IPv6 je podání žádosti u poskytovatele služeb o přiřazení IPv6 prefixu. Hned po přidělení prefixu /48 od ISP je rozvaha, jak solidně a přehledně naložit s tak velkým rozsahem a přitom dodržet základní charakteristiky IPv6. Takovou charakteristikou může být přidělování /64 bitového prefixu koncovým stanicím, dodržování autokonfigurace a další. Rozvahu pro síť si každý poskytovatel vypracovává sám dle svých interních požadavků. Jako ukázka rozdělení IPv6 rozsahu pro MAN síť je k této bakalářské práci přiložena příloha, která toto rozdělení prezentuje. Rozvaha počítá s tím, že je na síti každá adresa veřejná a zároveň při pohledu na adresu je možné říci, k jakému účelu na síti slouží. Dále je lépe vidět, kolik je adres k dispozici. K této rozvaze je dobré poznamenat, že se jedná o prvotní návrh, který se během praktického nasazení může dodatečně změnit. V IPv6 se jedná totiž o možnosti prakticky nasazovat

NPT, známý také jako NAT66. Tento NAT je definovaný v RFC 6296. Ano, opravdu je řeč o překladu IPv6 adres, i když je rozsah IPv6 nadměrný. Co ale případná readresace sítě, třeba v případě změny ISP? V takové situaci by bylo nutné v síti změnit všem prvků IP adresy, což by bylo dost nepříjemné. V takové situaci se nabízí varianta úvahy o nasazení adres místní stránky (`fc00::/7`), a na serveru NPT dělat překlad 1:1.

Výhody oproti předchozímu příkladu jsou značné. Případná readresace nezpůsobí velký problém a zároveň správce má k dispozici více adres pro nasazení na MAN síti. Další výhodou je, že na směrovačích nejsou veřejné adresy, protože jejich adresy místní stránky nebudou překládány, takže nemusí být na každém aplikována restrikce pro kontrolu provozu, nebo případných útoků. Tím jsou šetřeny systémové prostředky směrovače. [2]

6.1 Základní implementace

Po přidělení adresního rozsahu od ISP a rozvaze je možné začít IPv6 nasazovat. Nasazení bude demonstrováno na koncové stanici, protože z pohledu retranslační stanice platí pro IPv6 stejné principy omezování a řízení provozu jako pro IPv4. Je však možné, že se mohou vyskytnout případné rozdíly. Pak je tuto situaci nutné řešit individuálně. Je tedy k dispozici koncová stanice, které budou přiřazeny dva prefixy /64 pro dvě různá rozhraní. Tyto prefixy budou distribuovány do sítě automatickou konfigurací, která bude na koncové stanici nastavena. Dále tato koncová stanice musí být součástí dynamického směrování, aby všechny adresy definované na jejích rozhraních byly dostupné z celého světa. V nasazení jsou použity následující zkratky.

UNR - user network range TNR - transport network range

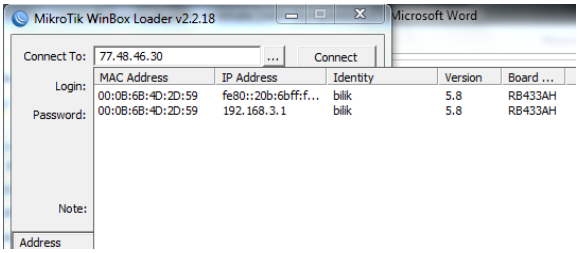
TNR: 2a01:5e0:b:ffff::0/124 => GW = 2a01….::1 ; klient = 2a01…..::2

DNS: 2a01:348::6:4d4b:69a9:0:1

UNR1: 2a01:5e0:b:8000/64

UNR2: 2a01:5e0:b:8001/64

Jako první budou v RouterOS přidána do Neighbor Discovery rozhraní, na kterých se bude IPv6 využívat. Po přidání je možné „aktivovat“ podporu autokonfigurace na tomto rozhraní. Důkaz o tom, že na tomto rozhraní IPv6 funguje, najdeme v utilitě Winbox. Po stisknutí tlačítka „třech teček“ je zobrazena u nastavovaného směrovače ne jen IPv4 adresa, ale i IPv6 s adresou pro lokální linku. Tento děj je zobrazen na obrázku 6.1.



Obr. 2 Zobrazení linkové IPv6 adresy v utilitě Winbox

Nyní je možné z operačního systému podporující IPv6 přistupovat k tomuto směrovači pomocí tohoto protokolu. Je možné si všimnout toho, že aniž by byla do RouterOS zadána jakákoiv IPv6 adresa, je automaticky nabídnuta. Toto je další vlastnost IPv6, automatické vygenerování linkové adresy, která se skládá z MAC adresy a počátku fe80, který je k tomuto účelu vyhrazen. Výhodu to má v tom, že již od samého počátku je možno do RouterOS přistupovat na stabilnějším připojením L3 vrstvy. Nyní budou přidána rozhraní „eth1“ a „vysilani2“ do ND

```
/ipv6 nd
add advertise-dns=yes advertise-mac-address=yes disabled=no \
    hop-limit=64 interface=ether1 managed-address-configuration=yes\
    mtu=1500 other-configuration=yes ra-delay=3s
   ra-interval=3m20s-10m ra-lifetime=30m reachable-time=unspecified\
   retransmit-interval=unspecified
add advertise-dns=yes advertise-mac-address=yes disabled=no \
    hop-limit=64 interface=vysilani2 ra-lifetime=30m \
    managed-address-configuration=yes mtu=unspecified \
    other-configuration=yes ra-delay=3s ra-interval=3m20s-10m \
    reachable-time=unspecified retransmit-interval=unspecified
```

Po absolvování tohoto kroku budou na dané rozhraní přidány IPv6 rozsahy.

```
/ipv6 address
add address=2a01:5e0:b:ffff::2/124 advertise=no comment=prijem\
    disabled=no eui-64=no interface=prijem

add address=2a01:5e0:b:8000:20c:42ff:fe28:6b2f/64 advertise=yes\
    comment="prefix /64 eth1" disabled=no eui-64=yes \
    interface=ether1
```

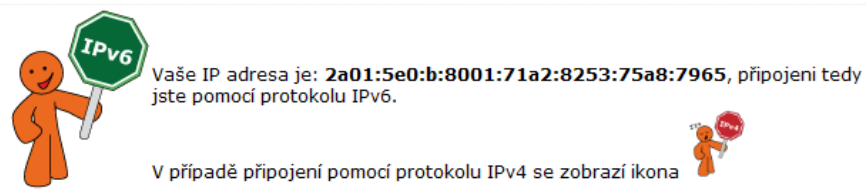
```
add address=2a01:5e0:b:8001:20b:6bff:fe4d:2d59/64 advertise=yes\
    comment="prefix /64 vysilani2" disabled=no eui-64=yes \
    interface=vysilani2
```

Parametr advertise musí být jako yes, protože tím je řečeno, že tento rozsah bude dostupný pro autokonfiguraci. Parametr EUI64 je funkce, která jednoznačně vypočítá z rozsahu /64 a MAC adresy rozhraní IPv6 adresu jednoznačnou v globálním měřítku. Tato funkce by se měla standardně používat. Jsou však situace, kdy se to nehodí, a tak zapnutá být nemusí, navíc při vedení dobré evidence použitých adres a rozsahů k duplicitě na síti nedojde. Pokud budou na rozhraní přidávány jiné rozsahy než /64, ani jeden parametr nebude možné zapnout, což je vidět v první ukázce, kde jsou definovány adresy příjmu. V takové situaci není možné použít autokonfiguraci, a to i kdyby se nejednalo o příjem. Jediná možnost je využití DHCP, které je od verze 5.9 v RouterOS dostupné. Nyní je potřeba nastavit dynamické směrování. Zásady pro nastavení jsou stejné, tak jak byly částečně popsány v kapitole 5. Jediná změna je, že místo IPv4 adresy je logicky IPv6 adresa. Je vhodné zakládat peer pro IPv6 zvlášť od peeru IPv4. Když vznikne chyba někde při nastavování peeru, nerozpadne se IP spojení úplně, ale jen toho protokolu, u kterého byla tato chyba způsobena. Takže bude-li chyba na páteřním peeru IPv6, spadne pouze spojení IPv6, ale přes IPv4 je tento směrovač stále dostupný a je možné bezpečně tuto chybu opravit. Navíc je možné nezávisle směrovat IPv4 provoz od IPv6 atd.

Následující nastavení je demonstrace nastavení peeru ze strany klienta.

```
add address-families=ipv6 as-override=no default-originate=always \
    disabled=no hold-time=3m in-filter="" instance=default \
    multihop=no name="ipv6 silo" nexthop-choice=default \
    out-filter="" passive=no remote-address= 2a01:5e0:b:ffff::1 \
    remote-as=60101 remove-private-as=no route-reflect=no
    tcp-md5-key="" ttl=default use-bfd=no
```

Tímto bylo dokončeno základní nastavení IPv6 protokolu a na stránce www.nix.cz jsme vítaní postavičkou se zelenou cedulkou. [2,6]



Obr. 3 Oznámení serveru nix.cz o připojení pomocí IPv6

7 Závěr

Cílem této bakalářské práce bylo analyzování operačního systému RouterOS, zda je vhodný pro využití v gigabitových sítích, uvést možnosti hardwarových architektur, které jsou s tímto operačním systémem kompatibilní a srovnat jejich výkon. Jako poslední bod, kterému se bakalářská práce věnuje, je reálná implementace RouterOS s využitím gigabitového rozhraní. Jsou zde prezentována základní síťová nastavení, která se mohou v běžné praxi objevovat. Byla uvedena také cela řada základních síťových pravidel popisujících chování sítě. Snahou bylo prezentovat taková síťová pravidla, která najdou využití nejen u velkých síťových poskytovatelů internetu, ale i v menších podnikových a domácích sítích.

Dle ukázek a praktických řešení můžeme říci, že společnost MikroTik nabízí vskutku profesionální a sofistikované řešení pro tvorbu telekomunikační datové sítě. I když z pohledu hardware má společnost MikroTik ještě co dohánět, z pohledu software se jedná o hojně vybavený, stabilní a cenově velice dostupný operační systém. V kombinaci s platformou x86, je možné výkon určit HW vybavením, tak lze získat univerzální systém, převážně pro poskytovatele datových služeb využívající technologii Wi-Fi. Velkou výhodou je také rychlá implementace mechanismů IPv6 protokolu, což znázorňuje jeho aktivní vývoj. Bude-li tak potřeba realizovat gigabitový prvek do sítě, je RouterOS, v kombinaci s x86, jednou z možných variant.

Z hlediska dalšího vývoje a předpokladu, že se do popředí bude dostávat protokol IPv6, a IPv4 bude postupně upadat, bude daleko citelnější potřeba chránit na veřejných adresách koncové uživatele a zároveň nabídnout výkonnou domácí multimediální gigabitovou síť. MikroTik se pak může stát skvělým řešením i v těchto požadavcích. K těmto účelům je možné využít síťová pravidla, která jsou použitá v ukázkách, nebo je případně rozšířit pro vlastní potřeby.

Přínosem této bakalářské práce jsou zmíněné zdrojové kódy, poukazující na tradiční problémy v sítích poskytovatelů a prezentace IPv6, jakož to nevyhnutelný krok v internetu budoucnosti.

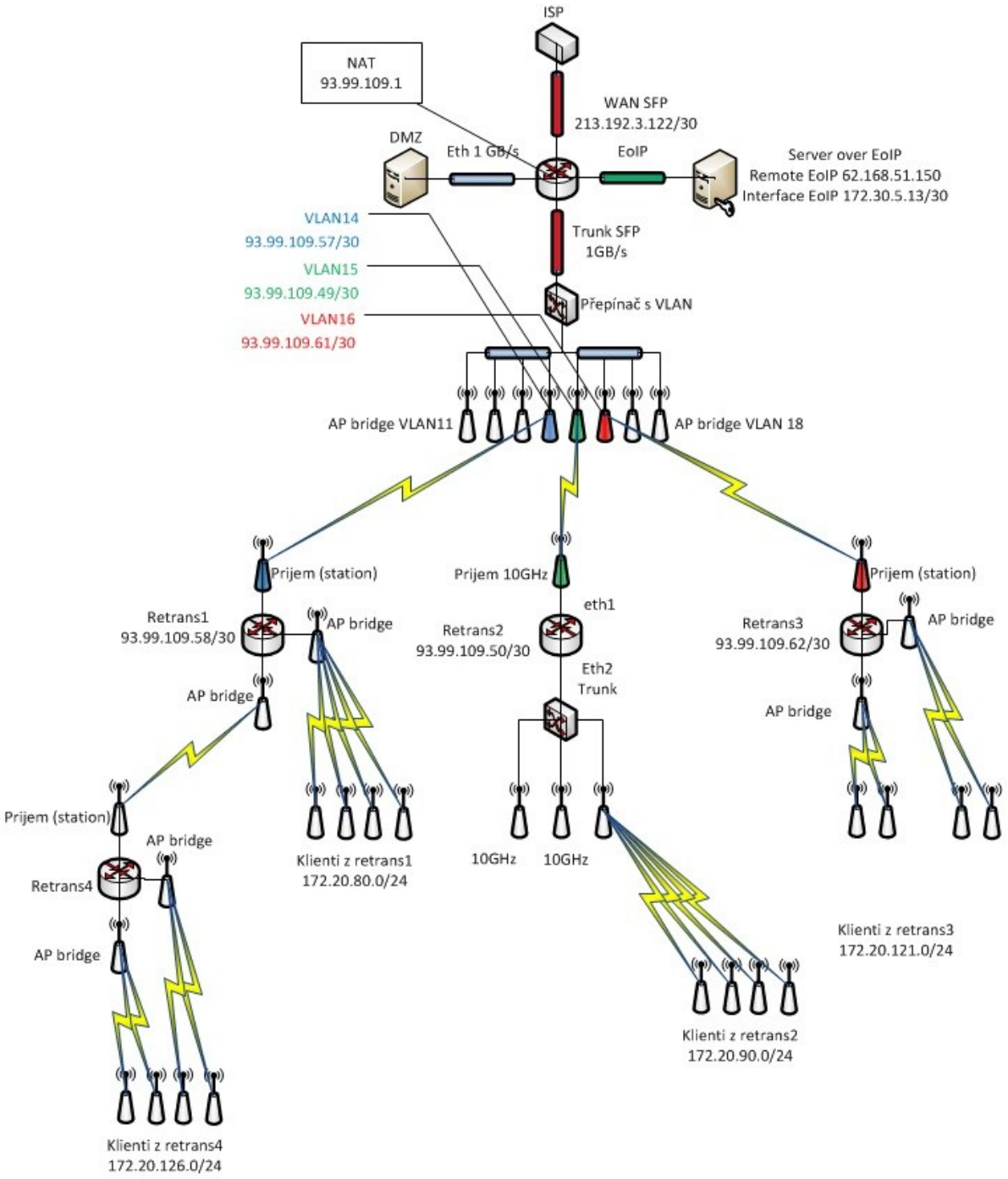
Použitá literatura

1. PUŽMANOVÁ, R. *Propojování sítí s TCP/IP*. České Budějovice: Kopp, 1999, 203 s. ISBN 80-723-2080-7.
2. SATRAPA, Pavel. *IPv6*: *internet protocol verze 6*. Vyd. 1. Praha: Neocortex, 2002, 238 s. ISBN 80-863-3010-9.
3. OHRTMAN, Frank. *Voice over 802.11*. Boston: Artech House, c2004, 258 s. ISBN 15-805-3677-8.
4. ODOM, Wendell, Rus HEALY a Naren MEHTA. *Směrování a přepínání sítí*: *autorizovaný výukový průvodce*. Vyd. 1. Brno: Computer Press, 2009, 879 s. ISBN 978-80-251-2520-5.
5. BARRETT, J. *SSH*: *kompletní průvodce*. Vyd. 1. Brno: SoftPress, 2003, 556 s. ISBN 80-7226-852-X (brož.).
6. *MikroTik Wiki* [online]. 1,0. Litva, 22.9.2011 [cit. 2012-04-22]. Dostupné z: wiki.mikrotik.com
7. *Wikipedia* [online]. 1,0. Czech Republic, 24.12.2010 [cit. 2012-04-20]. Dostupné z: http://cs.wikipedia.org/wiki/MikroTik_RouterOS

Seznam příloh

Příloha A

Základní schéma sítě



Příloha B

Ukázka adresního plánu IPv6 MAN sítě

Parametry od ISP:

Transportní adresa: 2a01:5e00::3:8/126

GW: 2a01:5e00::3:9

DNS: 2a01:348::6:4d4b:69a9:0:1

Přidělený rozsah: 2a01:5e0:B::/48

Hlavní úlohou rozdělení prefixu /48 na menší rozsahy je snadnější správa sítě a orientace v ní. Adresování je založeno na tom, že by každý účastník měl možnost mít k dispozici minimálně jeden /64 prefix. Výjimkou budou rozhraní páteřních linek, kde je určen prefix pro transporty /124.

Pro transport není úmyslně určen prefix /126 (čtyři adresy), protože jako nejmenší možnou část pro dělení jsem určil 4 bity (0-f), čím získáváme přehlednější dělení.

Návrh počítá s následujícími požadavky.

rozsah	účel	První adresa	Poslední adresa	Poznámka
2a01:5e0:b::/52	Firemní rozsahy	2a01:5e0:b::	2a01:5e0:b:0fff:ff..f	256x prefix /60
2a01:5e0:b:1::/52	Firemní rozsahy	2a01:5e0:b:1::	2a01:5e0:b:1fff:ff..f	256x prefix /60
2a01:5e0:b:2:: - 2a01:5e0:b:fffe::	Rozsah pro klienty	2a01:5e0:b:2::	2a01:5e0:b:fffe:ff..f	57 344x prefix /64
2a01:5e0:b:ffff::/64	Transportní adresy	2a01:5e0:b:ffff::	2a01:5e0:b:ffff:ff..f	moc

a) s možností přidělit účastníkovi více /64 prefixů

b) stanovit tvar a rozsah pro transportní adresy mezi páteřními uzly

c) stanovit tvar a rozsah pro transportní adresy mezi uzlem a uživatelem

d) stanovit rozsah, definující počet použitelných /64 prefixů

Pro firmy je stanovený primární prefix 60, což přiděluje 16 adres s prefixem 64, které se dají použít pro potřeby firmy. Celkem je k dispozici 512 prefixů 60 (prefix 52 = 256x prefix 60).

Transportní adresy se dají rozdělit do dalších podskupin, aby byly jednoduše identifikovatelné, které jsou pro transport mezi technologií a které jsou pro transport mezi klientem. Prefixem bude /124

2a01:5e0:B:ffff:AAAA:BCCC:DDDD:EEEE/124

kde číslo:

AAAA - určuje uzel, kde se spoj nachází

B - identifikuje transport mezi uzly nebo klienty

(**B**=0 – jedná o transport mezi uzly, **B**= 1- jedná se o transport mezi klientem a uzlem, 2-f reserva)

CCC – další, detailnější identifikace transportu, nebo přidání k rozsahu transportních adres a čísla D a E rozsah pro transportní adresy

2a01:5e0:b:BCCx:xxxx:xxxx:xxxx:xxxx/60 - **Rozsah pro firmy**

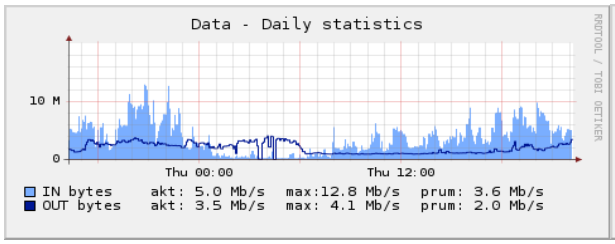
B=0,1 => CC číslo prefixu /60 Př:2a01:5e0:b:1020::/60 ; 2a01:5e0:b:10f0::/60

2a01:5e0:b:BCCC:xxxx:xxxx:xxxx:xxxx/64 **Rozsah pro klienty**

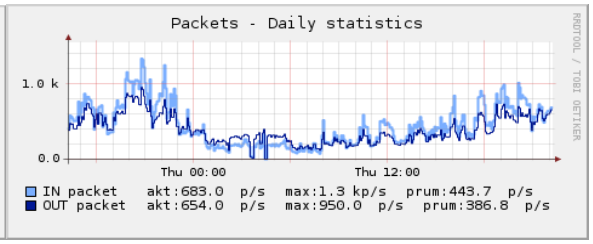
B>1 +CCC pořadí klientského prefixu /64

Příloha C

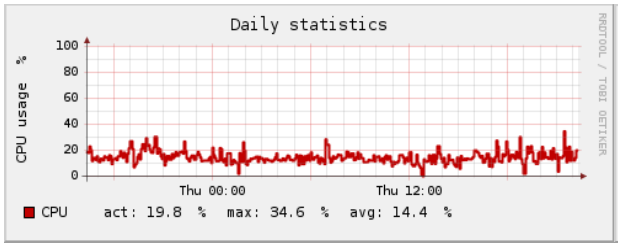
Grafy naměřených hodnot architektury x86



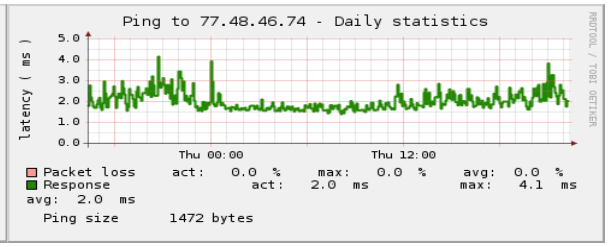
Obr. 2.1 Datový průtok architekturou x86



Obr. 2.2 Paketový průtok architekturou x86

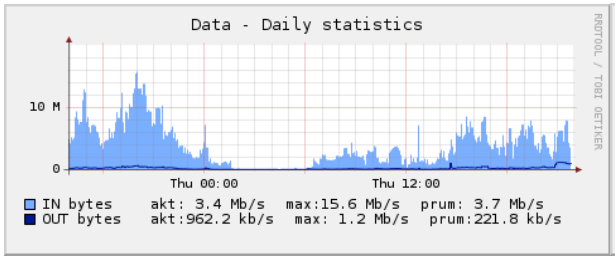


Obr. 2.3 Využití CPU architekturou x86

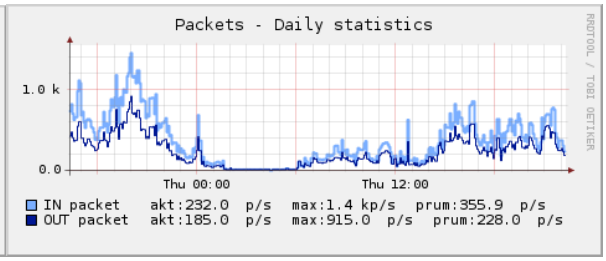


Obr. 2.4 Latence retranslační stanice

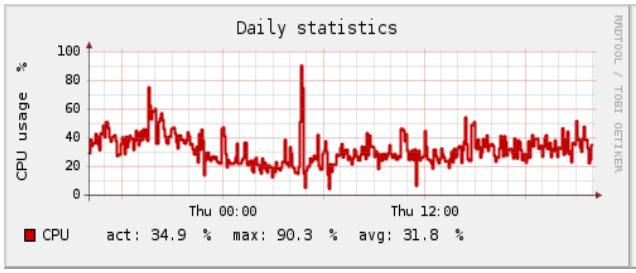
Grafy naměřených hodnot architektury PowerPC



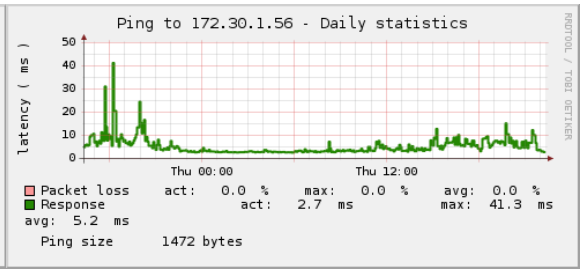
Obr. 2.5 Datový průtok architekturou PowerPC



Obr. 2.6 Paketový průtok architekturou PowerPC



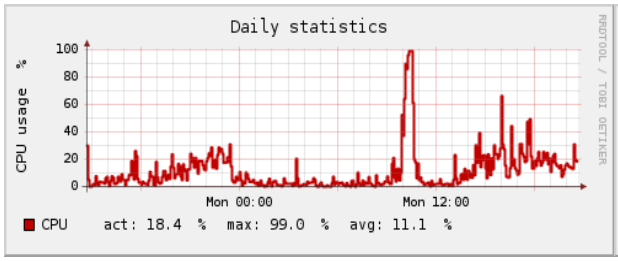
Obr. 2.7 Využití CPU architekturou PowerPC



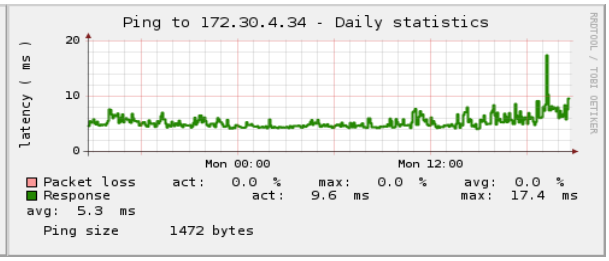
Obr. 2.8 Latence retranslační stanice PowerPC

Příloha D

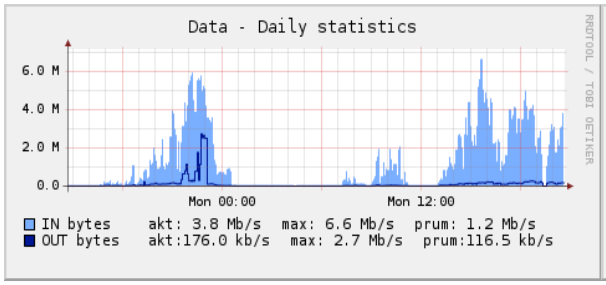
Grafy naměřených hodnot architektury MIPSbe



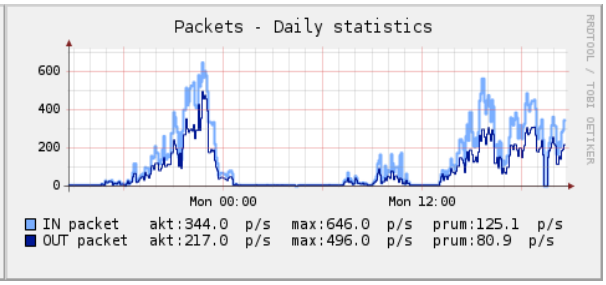
Obr. 4 Využití CPU architekturou MIPSbe



Obr. 5 Latence na retranslační stanici

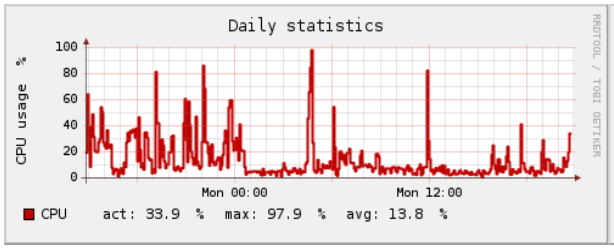


Obr. 6 Datový průtok architekturou MIPSbe

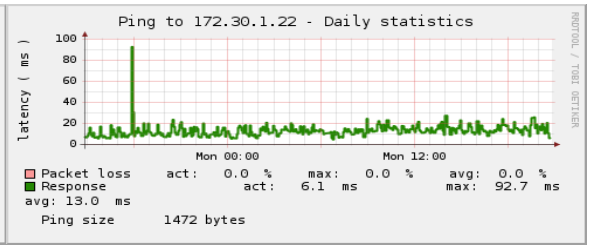


Obr. 7 Paketový průtok architekturou MIPSbe

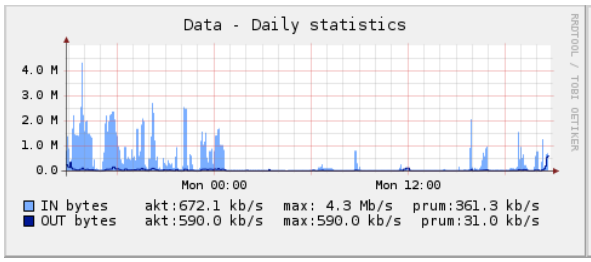
Grafy naměřených hodnot architektury MIPSle



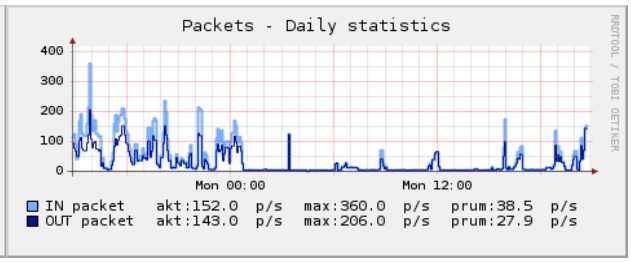
Obr. 8 Využití CPU architekturou MIPSle



Obr. 9 Latence na retranslační stanici



Obr. 10 Datový průtok architekturou MIPSle



Obr. 11 Paketový průtok architekturou MIPSle